

# Neural Rendering Model: Joint Generation and Prediction for Semi-Supervised Learning

Nhat Ho<sup>\*,◊</sup>, Tan Nguyen<sup>†,◊</sup>, Ankit Patel<sup>‡,‡</sup>,  
Anima Anandkumar<sup>††,‡,◊</sup>, Michael I. Jordan<sup>\*,◊</sup>, Richard G. Baraniuk<sup>†,◊</sup>

University of California at Berkeley, Berkeley, USA <sup>\*</sup>

Rice University, Houston, USA <sup>†</sup>

Baylor College of Medicine, Houston, USA <sup>‡</sup>

California Institute of Technology, Pasadena, USA <sup>††</sup>

NVIDIA, Santa Clara, USA <sup>‡‡</sup>

November 8, 2018

## Abstract

Unsupervised and semi-supervised learning are important problems that are especially challenging with complex data like natural images. Progress on these problems would accelerate if we had access to appropriate generative models under which to pose the associated inference tasks. Inspired by the success of Convolutional Neural Networks (CNNs) for supervised prediction in images, we design the Neural Rendering Model (NRM), a new probabilistic generative model whose inference calculations correspond to those in a given CNN architecture. The NRM uses the given CNN to design the prior distribution in the probabilistic model. Furthermore, the NRM generates images from coarse to finer scales. It introduces a small set of latent variables at each level, and enforces dependencies among all the latent variables via a conjugate prior distribution. This conjugate prior yields a new regularizer based on paths rendered in the generative model for training CNNs—the Rendering Path Normalization (RPN). We demonstrate that this regularizer improves generalization, both in theory and in practice. In addition, likelihood estimation in the NRM yields training losses for CNNs, and inspired by this, we design a new loss termed as the Max-Min cross entropy which outperforms the traditional cross-entropy loss for object classification. The Max-Min cross entropy suggests a new deep network architecture, namely the Max-Min network, which can learn from less labeled data while maintaining good prediction performance. Our experiments demonstrate that the NRM with the RPN and the Max-Min architecture exceeds or matches the-state-of-art on benchmarks including SVHN, CIFAR10, and CIFAR100 for semi-supervised and supervised learning tasks.

**Keywords:** neural nets, generative models, semi-supervised learning, cross-entropy, statistical guarantee

## 1 Introduction

Unsupervised and semi-supervised learning have still lagged behind compared to performance leaps we have seen in supervised learning over the last five years. This is partly due to a lack of good generative models that can capture all latent variations in complex domains such as natural images and provide useful structures that help learning. When it comes to probabilistic generative models, it is hard to design good priors for the latent variables that drive the generation.

---

◊ Nhat Ho and Tan Nguyen contributed equally to this work.

◊ Anima Anandkumar, Michael I. Jordan, and Richard G. Baraniuk contributed equally to this work.

Instead, recent approaches avoid the explicit design of image priors. For instance, Generative Adversarial Networks (GANs) use implicit feedback from an additional discriminator that distinguishes real from fake images [1]. Using such feedback helps GANs to generate visually realistic images, but it is not clear if this is the most effective form of feedback for predictive tasks. Moreover, due to separation of generation and discrimination in GANs, there are typically more parameters to train, and this might make it harder to obtain gains for semi-supervised learning in the low (labeled) sample setting.

We propose an alternative approach to GANs by designing a class of probabilistic generative models, such that inference in those models also has good performance on predictive tasks. This approach is well-suited for semi-supervised learning since it eliminates the need for a separate prediction network. Specifically, we answer the following question: what generative processes output Convolutional Neural Networks (CNNs) when inference is carried out? This is natural to ask since CNNs are state-of-the-art (SOTA) predictive models for images, and intuitively, such powerful predictive models should capture some essence of image generation. However, standard CNNs are not directly reversible and likely do not have all the information for generation since they are trained for predictive tasks such as image classification. We can instead invert the irreversible operations in CNNs, e.g., the rectified linear units (ReLUs) and spatial pooling, by assigning auxiliary latent variables to account for uncertainty in the CNN’s inversion process due to the information loss.

**Contribution 1 – Neural Rendering Model:** We develop the Neural Rendering Model (NRM) whose bottom-up inference corresponds to a CNN architecture of choice (see Figure 1a). The “reverse” top-down process of image generation is through coarse-to-fine rendering, which progressively increases the resolution of the rendered image (see Figure 1b). This is intuitive since the reverse process of bottom-up inference reduces the resolution (and dimension) through operations such as spatial pooling. We also introduce structured stochasticity in the rendering process through a small set of discrete latent variables, which capture the uncertainty in reversing the CNN feed-forward process. The rendering in NRM follows a product of linear transformations, which can be considered as the transpose of the inference process in CNNs. In particular, the rendering weights in NRM are proportional to the transpose of the filters in CNNs. Furthermore, the bias terms in the ReLU units at each layer (after the convolution operator) make the latent variables in different network layers dependent (when the bias terms are non-zero). This design of image prior has an interesting interpretation from a predictive-coding perspective in neuroscience: the dependency between latent variables can be considered as a form of backward connections that captures prior knowledge from coarser levels in NRM and helps adjust the estimation at the finer levels [2, 3]. The correspondence between NRM and CNN is given in Figure 2 and Table 1 below.

NRM is a likelihood-based framework, where unsupervised learning can be derived by maximizing the expected complete-data log-likelihood of the model while supervised learning is done through optimizing the class-conditional log-likelihood. Semi-supervised learning unifies both log-likelihoods into an objective cost for learning from both labeled and unlabeled data. The NRM prior has the desirable property of being a conjugate prior, which makes learning in NRM computationally efficient.

Interestingly, we derive the popular *cross-entropy loss* used to train CNNs for supervised learning *as an upper bound of the NRM’s negative class-conditional log-likelihood*. A broadly-accepted interpretation of cross-entropy loss in training CNNs is from logistic regression perspective. Given features extracted from the data by the network, logistic regression is applied to classify those features into different classes, which yields the cross-entropy. In this interpretation, there is a gap

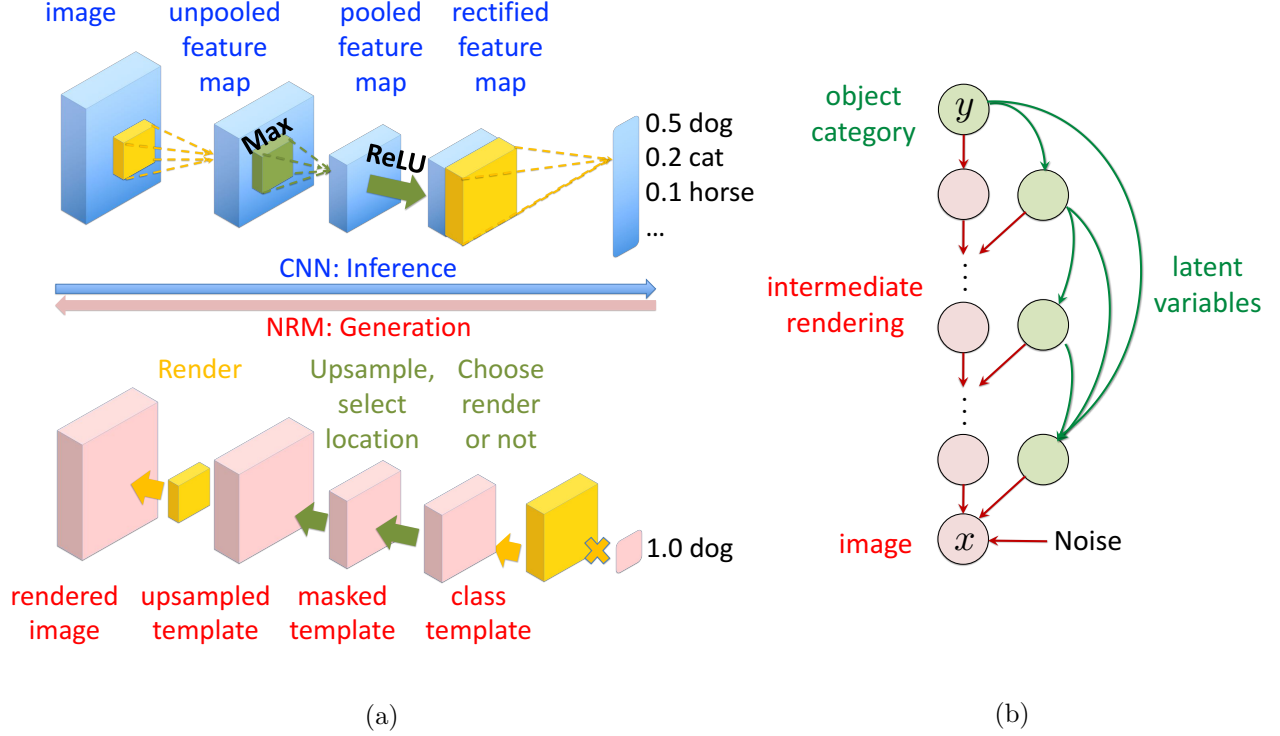


Figure 1: (a) *Neural Rendering Model (NRM)* is a probabilistic generative model that captures latent variations in the data and reduces CNNs as its inference algorithm for supervised learning. In particular, CNN architectures and training losses can be derived from a joint maximum a posteriori (JMAP) inference and likelihood estimation in NRM, respectively. NRM can learn from unlabeled data, overcoming the weakness of CNNs, and achieve good performance on various semi-supervised learning tasks. (b) *Graphical model depiction of NRM*. Latent variables in NRM depend on each other. Starting from the top of the model with the object category  $y$ , new latent variables are incorporated into the model at each layer, and intermediate images are rendered with finer details. At the bottom of NRM, pixel noise is added to render the final image  $x$ .

between feature extraction and classification. On the contrary, our derivation ties feature extraction and learning for classification in CNN into an end-to-end optimization problem that estimates the conditional log-likelihood of NRM. This new interpretation of cross-entropy allow us to develop better losses for training CNNs. An example is the *Max-Min cross-entropy* discussed in Contribution 2 and Section 5.

**Contribution 2 – New regularization, loss function, architecture and generalization bounds:** The joint nature of generation, inference, and learning in NRM allows us to develop new training procedures for semi-supervised and supervised learning, as well as new theoretical (statistical) guarantees for learning. In particular, for training, we derive a new form of regularization termed as the *Rendering Path Normalization (RPN)* from the NRM’s conjugate prior. A rendering path is a set of latent variable values in NRM. Unlike the path-wise regularizer in [4], RPN uses information from a generative model to penalizes the number of the possible rendering paths and, therefore, encourages the network to be compact in terms of representing the image. It also helps to enforce the dependency among different layers in NRM during training and improves classification

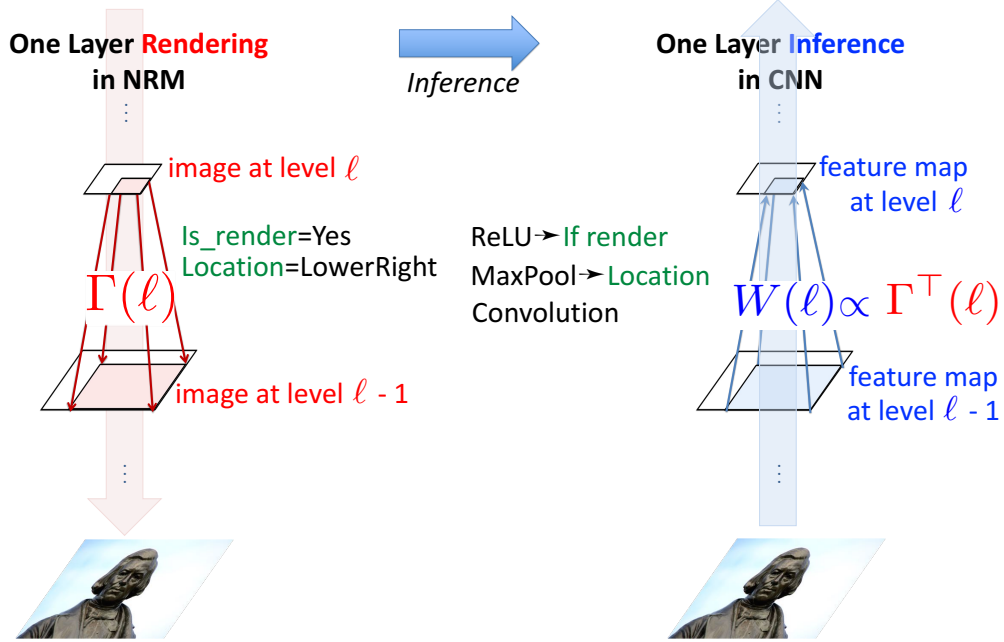


Figure 2: CNN is the E-step JMAP inference of the optimal latent variable  $z(\ell)$  in NRM. In particular, inferring the template selecting latent variables  $s(\ell)$  in NRM results in the ReLU non-linearities in CNN. Similarly, inferring the local translation latent variables  $t(\ell)$  and inverting the zero-padding in NRM yield the MaxPool operators in CNN. In addition, during inference, the rendering step using the templates  $\Gamma(\ell)$  in NRM becomes the convolutions with weights  $W(\ell) = \Gamma^T(\ell)$  in the CNN.

Table 1: Correspondence between NRM and CNN

NRM	CNN
Rendering templates $\Gamma(\ell)$	Transpose of filter weights $W(\ell)$
Class templates $\mu(y)$	Softmax weights
Parameters $b(t; \ell)$ of the conjugate prior $\pi_{y x}$	Bias terms $b(\ell)$ in the ReLUs after each convolution
Max-marginalize over $s(\ell)$	ReLU
Max-marginalize over $t(\ell)$	MaxPool
Conditional log-likelihood $\frac{1}{n} \sum_{i=1}^n \log p(y_i   x_i, z_i; \theta)$	Cross-entropy loss
Expected complete-data log-likelihood $\frac{1}{n} \sum_{i=1}^n \mathbb{E} [\log p(x_i, z_i   y_i)]$	Reconstruction loss
Normalize the intermediate image $h(\ell)$	Batch Normalization

performance.

We provide new theoretical bounds based on NRM. In particular, we prove that NRM is statistically consistent and derive a generalization bound of NRM for (semi-)supervised learning tasks. Our generalization bound is proportional to the number of active rendering paths that generate close-to-real images. This suggests that RPN regularization may help in generalization since RPN enforces the dependencies among latent variables in NRM and, therefore, reduces the number of active rendering paths. We observe that RPN helps improve generalization in our experiments.

*Max-Min cross-entropy and network:* We propose the new *Max-Min cross-entropy* loss function for learning, based on negative class-conditional log-likelihood in NRM. It combines the traditional cross-entropy with another loss, which we term as the Min cross-entropy. While the traditional (Max) cross-entropy maximizes the probability of the correct labels, the Min cross-entropy minimizes the probability of the incorrect labels. We show that the Max-Min cross-entropy is also an upper bound to the negative conditional log-likelihood of NRM, just like the cross-entropy loss. The Max-Min cross-entropy is realized through a new CNN architecture, namely the *Max-Min network*, which is a CNN with an additional branch sharing weights with the original CNN but containing minimum pooling (MinPool) operator and negative rectified linear units (NReLU), i.e.,  $\min(\cdot, 0)$  (see Figure 5). Although the Max-Min network is derived from NRM, it is a meta-architecture that can be applied independently on any CNN architecture. We show empirically that Max-Min networks and cross-entropy help improve the SOTA on object classification for supervised and semi-supervised learning.

**Contribution 3 – State-of-the-art empirical results for semi-supervised and supervised learning:** We show strong results for semi-supervised learning over CIFAR10, CIFAR100 and SVHN benchmarks in comparison with SOTA methods that use and do not use consistency regularization. Consistency regularization, such as those used in Temporal Ensembling [5] and Mean Teacher [6], enforces the networks to learn representation invariant to realistic perturbations of the data. NRM alone outperforms most SOTA methods which do not use consistency regularization [7, 8] in most settings. Max-Min cross-entropy then helps improve NRM’s semi-supervised learning results significantly. When combining NRM, Max-Min cross-entropy, and Mean Teacher, we achieve SOTA results or very close to those on CIFAR10, CIFAR100, and SVHN (see Table 3, 4, and 5). Interestingly, compared to the other competitors, our method is consistently good, achieving either best or second best results in all experiments. Furthermore, Max-Min cross-entropy also helps supervised learning. Using the Max-Min cross-entropy, we achieve SOTA result for supervised learning on CIFAR10 (2.30% test error). Similarly, Max-Min cross-entropy helps improve supervised training on ImageNet.

Despite good classification results, there is a caveat that NRM may not generate good looking images since that objective is not “baked” into its training. NRM is primarily aimed at improving semi-supervised and supervised learning through better regularization. Potentially, an adversarial loss can be added to NRM to improve visual characteristics of the image, but that is beyond the scope of this paper.

**Notation:** To facilitate the presentation, the NRM’s notations are explained in Table 2. Throughout this paper, we denote  $\|x\|$  and  $x^\top$  the Euclidean norm and transpose of  $x$  respectively for any  $x \in \mathbb{R}^d$ . Additionally, for any matrix  $A$  and  $B$  of the same dimension,  $A \odot B$  denotes the Hadamard product between  $A$  and  $B$ . For any vector  $a$  and  $b$  of the same dimension,  $\langle a, b \rangle$  denotes the dot product between  $a$  and  $b$ .

Table 2: Table of notations for NRM

<u>Variables</u>	
$x$	$\triangleq$ input image of size $D(0)$
$y$	$\triangleq$ object category
$z(\ell) = \{s(\ell), t(\ell)\}$	$\triangleq$ all latent variables of size $D(\ell)$ in layer $\ell$
$s(\ell, p)$	$\triangleq$ switching latent variable at pixel location $p$ in layer $\ell$
$t(\ell, p)$	$\triangleq$ local translation latent variable at pixel location $p$ in layer $\ell$
$h(y, z; \ell) = h(\ell)$	$\triangleq$ intermediate rendered image of size $D(\ell)$ in layer $\ell$
$h(y, z; 0) = h(0)$	$\triangleq$ rendered image of size $D(0)$ from NRM before adding noise
$\psi(\ell)$	$\triangleq$ corresponding feature maps in layer $\ell$ in CNNs.
<u>Parameters</u>	
$\mu(y) = h(y; L) = h(L)$	$\triangleq$ the template of class $y$ , as well as the coarsest image of size $D(L)$ determined by the category $y$ at the top of NRM before adding any fine detail. $\mu(y)$ is learned from the data.
$\Lambda(\ell)$	$\triangleq$ rendering matrix of size $D(\ell - 1) \times D(\ell)$ at layer $\ell$ .
$\Gamma(\ell)$	$\triangleq$ dictionary of $D(\ell)$ rendering template $\Gamma(\ell, p)$ of size $F(\ell) \times 1$ at layer $\ell$ . $\Gamma(\ell)$ is learned from the data.
$W(\ell) = \Gamma^\top(\ell, p)$	$\triangleq$ corresponding weight at the layer $\ell$ in CNNs
$B(\ell)$	$\triangleq$ set of zero-padding matrices $B(\ell, p) \in \mathbb{R}^{D(\ell-1) \times F(\ell)}$ at layer $\ell$
$T(\ell)$	$\triangleq$ set of local translation matrices $T(\ell, p) \in \mathbb{R}^{D(\ell-1) \times D(\ell-1)}$ at layer $\ell$ . $T(\ell, p)$ is chosen according to value of $t(\ell, p)$
$b(t; \ell) = b(\ell)$	$\triangleq$ parameter of the conjugate prior $p(z x, y)$ at layer $\ell$ . This term is of size $D(\ell)$ and becomes the bias term after convolutions in CNNs. It can be made independent of $t$ , which is equivalent to using the same bias in each feature map in CNNs. Here, $b(t; \ell)$ is learned from data.
$\pi_y$	$\triangleq$ probability of object category $y$ .
$\sigma^2$	$\triangleq$ pixel noise variance
<u>Other Notations</u>	
$\eta(y, z)$	$\triangleq \sum_{\ell=1}^L \frac{1}{\sigma^2} \langle b(t; \ell), s(\ell) \odot h(\ell) \rangle.$
Softmax ( $\eta$ )	$\triangleq \frac{\exp(\eta)}{\sum_{\eta'} \exp(\eta')}.$
RPN	$\triangleq -\frac{1}{n} \sum_{i=1}^n \log p(z_i^*   y_i) = -\frac{1}{n} \sum_{i=1}^n \text{Softmax}(\eta(y_i, z_i^*)).$
(y,z(L), ..., z(1))	$\triangleq$ rendering configuration.

## 2 Related Work

**Deep Generative Models:** In addition to GANs, other recently developed deep generative models include the Variational Autoencoders (VAE) [9] and the Deep Generative Networks [10]. Unlike these models, which replace complicated or intractable inference by CNNs, NRM derives CNNs as its inference. This advantage allows us to develop better learning algorithms for CNNs with statistical guarantees, as being discussed in Section 3.3 and 4. Recent works including the Bidirectional GANs [11] and the Adversarially Learned Inference model [8] try to make the discriminators and generators in GANs reversible of each other, thereby providing an alternative way to invert CNNs. These approaches, nevertheless, still employ a separate network to bypass the irreversible operators in CNNs. Furthermore, the flow-based generative models such as NICE [12], Real NVP [13], and Glow [14] are invertible. However, the inference algorithms of these models, although being exact, do not match the CNN architecture. NRM is also close in spirit to the Deep Rendering Model (DRM) [15] but markedly different. Compared to NRM, DRM has several limitations. In particular, all the latent variables in DRM are assumed to be independent, which is rather unrealistic. This lack of dependency causes the missing of the bias terms in the ReLUs of the CNN derived from DRM. Furthermore, the cross-entropy loss used in training CNNs for supervised learning tasks is not captured naturally by DRM. Due to these limitations, model consistency and generalization bounds are not derived for DRM.

**Semi-Supervised Learning:** In addition to deep generative model approach, consistency regularization methods, such as Temporal Ensembling [5] and Mean Teacher [6], have been recently developed for semi-supervised learning and achieved state-of-the-art results. These methods enforce that the baseline network learns invariant representations of the data under different realistic perturbations. Consistency regularization approaches are complimentary to and can be applied on most deep generative models, including NRM, to further increase the baseline model’s performance on semi-supervised learning tasks. Experiments in Section 6 demonstrate that NRM achieves better test accuracy on CIFAR10 and CIFAR100 when combined with Mean Teacher.

**Explaining Architecture of CNNs:** The architectures and training losses of CNNs have been studied from other perspectives. [16, 17] employ principles from information theory such as the Information Bottleneck Lagrangian introduced by [18] to show that stacking layers encourages CNNs to learn representations invariant to latent variations. They also study the cross-entropy loss to understand possible causes of over-fitting in CNNs and suggest a new regularization term for training that helps the trained CNNs generalize better. This regularization term relates to the amount of information about the labels memorized in the weights. Additionally, [19] suggests a connection between CNNs and the convolutional sparse coding (CSC) [20, 21, 22, 23]. They propose a multi-layer CSC (ML-CSC) model and prove that CNNs are the thresholding pursuit serving the ML-CSC model. This thresholding pursuit framework implies alternatives to CNNs, which is related to the deconvolutional and recurrent networks. The architecture of CNNs is also investigated using the wavelet scattering transform. In particular, scattering operators help elucidate different properties of CNNs, including how the image sparsity and geometry captured by the networks [24, 25]. In addition, CNNs are also studied from the optimization perspective [26, 27, 28, 29], statistical learning theory perspective [30, 31], approximation theory perspective [32], and other approaches [33, 34].

Like these works, NRM helps explain different components in CNNs. It employs tools and methods in probabilistic inference to interpret CNN from a probabilistic perspective. That says, NRM can potentially be combined with aforementioned approaches to gain a better understanding of CNNs.

### 3 The Neural Rendering Model

We first define the Neural Rendering Model (NRM). Then we discuss the inference in NRM. Finally, we derive different learning losses from NRM, including the cross-entropy loss, the reconstruction loss, and the RPN regularization, for supervised learning, unsupervised learning, and semi-supervised learning.

#### 3.1 Generative Model

NRM attempts to invert CNNs as its inference so that the information in the posterior  $p(y|x)$  can be used to inform the generation process of the model. NRM realizes this inversion by employing the structure of its latent variables. Furthermore, the joint prior distribution of latent variables in the model is parametrized such that it is the conjugate prior to the likelihood of the model. This conjugate prior is the function of intermediate rendered images in NRM and implicitly captures the dependencies among latent variables in the model. More precisely, NRM can be defined as follows:

**Definition 3.1. [Neural Rendering Model (NRM)]** NRM is a deep generative model in which the latent variables  $z(\ell) = \{t(\ell), s(\ell)\}_{\ell=1}^L$  at different layers  $\ell$  are dependent. Let  $x$  be the input image and  $y \in \{1, \dots, K\}$  be the target variable, e.g. object category. Generation in NRM takes the form:

$$\pi_{z|y} \triangleq \text{Softmax} \left( \frac{1}{\sigma^2} \eta(y, z) \right) \quad (1)$$

$$z|y \sim \text{Cat}(\pi_{z|y})$$

$$h(y, z; 0) \triangleq \Lambda(z; 1) \Lambda(z; 2) \cdots \Lambda(z; L) \mu(y) \quad (2)$$

$$x|z, y \sim \mathcal{N}(h(0), \sigma^2 \mathbf{1}_{D(0)}),$$

where:

$$\eta(y, z) \triangleq \sum_{\ell=1}^L \langle b(t; \ell), s(\ell) \odot h(\ell) \rangle \quad (3)$$

$$\text{Softmax}(\eta) \triangleq \frac{\exp(\eta)}{\sum_{\eta} \exp(\eta)}. \quad (4)$$

The generation process in NRM can be summarized in the following steps:

1. Given the class label  $y$ , NRM first samples the latent variables  $z$  from a categorical distribution whose prior is  $\pi_{z|y}$ .
2. Starting from the class label  $y$  at the top layer  $L$  of the model, NRM renders its coarsest image,  $h(L) = \mu(y)$ , which is also the object template of class  $y$ .



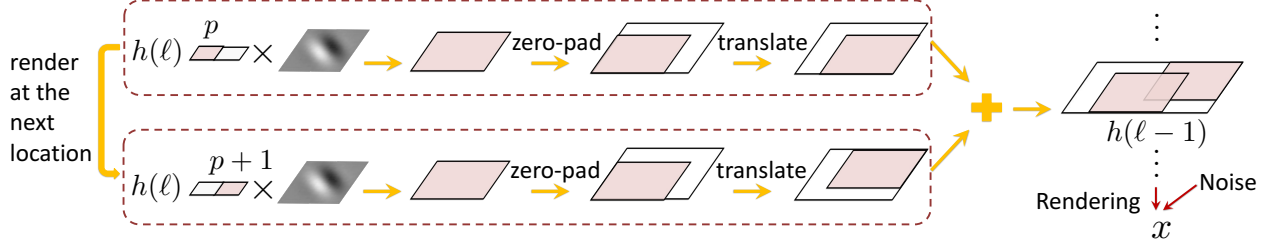


Figure 3: Rendering process from layer  $\ell$  to layer  $\ell - 1$  in NRM. At each pixel  $p$  in the intermediate image  $h(\ell)$ , NRM decides to render if the template selecting latent variable  $s(\ell, p) = 1$ . Otherwise, it does not render. If rendering, the template  $\Gamma(\ell, p)$  is multiplied by the pixel value  $h(\ell, p)$ . Then the matrix  $B(\ell, p)$  zero-pads the result to the size of the intermediate image at layer  $\ell - 1$ , which is  $D(\ell - 1) \times D(\ell - 1)$ . After that, the translation matrix  $T(\ell, p)$  locally translates the rendered image to location specified by the latent variable  $t(\ell, p)$ . The same rendering process repeats at pixel  $p + 1$ , as well as at other pixels of  $h(\ell)$ . NRM adds all rendered images to achieve the intermediate image  $h(\ell - 1)$  at layer  $\ell - 1$ .

3. At layer  $L - 1$ , a set of latent variations  $z(L)$  is incorporated into  $h(y; L)$  via a linear transformation  $\Lambda(z; L)$  to render the finer image  $h(y, z; L - 1)$ . The same process is repeated at each subsequent layer  $\ell$  to finally render the finest image  $h(y, z; 0)$  at the bottom of NRM.
4. Gaussian pixel noise is added into  $h(y, z; 0)$  to render the final image  $x$ .

In the generation process above,  $\Lambda(z; \ell)$  can be any linear transformation, and the latent variables  $z(\ell)$ , in their most generic form, can capture any latent variation in the data. While such generative model can represent any possible imagery data, it cannot be learned to capture the properties of natural images in reasonable time due to its huge number of degrees of freedom. Therefore, it is necessary to have more structures in NRM given our prior knowledge of natural images. One such prior knowledge is from classification models,  $p(y|x)$ . In particular, since classification models like CNNs have achieved excellent performance on a wide range of object classification tasks, we hypothesize that a generative model whose inference yields CNNs will also be a good model for natural images. As a result, we would like to introduce new structures into NRM so that CNNs can be derived as NRM’s inference. In other word, we use the posterior  $p(y|x)$ , i.e., CNNs, to inform the likelihood  $p(x|y)$  in designing NRM.

In our attempt to invert CNNs, we constrain the latent variables  $z(\ell)$  at layer  $\ell$  in NRM to a set of template selecting latent variables  $s(\ell)$  and local translation latent variables  $t(\ell)$ . As been shown later in Section 3.2, during inference of NRM, the ReLU non-linearity at layer  $\ell$  “inverts”  $s(\ell)$  to find if particular features are in the image or not. Similarly, the MaxPool operator “inverts”  $t(\ell)$  to locate where particular features, if exist, are in the image. Both  $s(\ell)$  and  $t(\ell)$  are vectors indexed by  $p(\ell) \in \mathcal{P}(\ell) \equiv \{\text{pixels in layer } \ell\}$ .

The rendering matrix  $\Lambda(\ell)$  is now a function of  $s(\ell)$  and  $t(\ell)$ , and the rendering process from layer  $\ell$  to layer  $\ell - 1$  using  $\Lambda(s, t; \ell)$  is described in the following equation:

$$h(\ell - 1) \triangleq \Lambda(\ell)h(\ell) = \sum_{p \in \mathcal{P}(\ell)} s(\ell, p)T(t; \ell, p)B(\ell, p)\Gamma(\ell, p)h(\ell, p). \quad (5)$$

Even though the rendering equation above seems complicated at first, it is quite intuitive as illustrated in Figure 3. At each pixel  $p$  in the intermediate image  $h(\ell)$  at layer  $\ell$ , NRM decides to

use that pixel to render or not according to the value of the template selecting latent variable  $s(p; \ell)$  at that pixel location. If  $s(p; \ell) = 1$ , then NRM renders. Otherwise, it does not. If rendering, then the pixel value  $h(\ell, p)$  is used to scale the rendering template  $\Gamma(\ell, p)$ . This rendering template is local. It has the same number of feature maps as the next rendered image  $h(\ell - 1)$ , but is of smaller size, e.g.  $3 \times 3$  or  $5 \times 5$ . As a result, the rendered image corresponds to a local patch in  $h(\ell - 1)$ . Next, the padding matrix  $B(\ell, p)$  pads the resultant patch to the size of the image  $h(\ell - 1)$  with zeros, and the translation matrix  $T(t; \ell, p)$  translates the result to a local location. NRM then keeps rendering at other pixel location of  $h(\ell)$  following the same process. All rendered images are added to form the final rendered image  $h(\ell - 1)$  at layer  $\ell - 1$ .

Note that in NRM, there is one rendering template at each pixel location  $p$  in the image  $h(\ell)$ . For example, if  $h(\ell)$  is of size  $128 \times 6 \times 6$ , then the NRM uses 4608 templates to render at layer  $\ell$ . This is too many rendering templates and would require a very large amount of data to learn, considering all layers in NRM. Therefore, we further constrain NRM by enforcing all pixels in the same feature maps of  $h(\ell)$  share the same rendering template. In other word,  $\Gamma(\ell, p)$  are the same if  $p$  are in the same feature map. This constrain helps yield convolutions in CNNs during the inference of NRM, and the rendering templates in NRM now correspond to the convolution filters in CNNs.

While  $s(\ell)$  and  $t(\ell)$  can be let independent, we further constrain the model by *enforcing the dependency among  $s(\ell)$  and  $t(\ell)$  at different layers in NRM*. This constraint is motivated from realistic rendering of natural objects: *different parts of a natural object are dependent on each other*. For example, in an image of a person, the locations of the eyes in the image are restricted by the location of the head or if the face is not painted, then it is likely that we cannot find the eyes in the image either. Thus, NRM tries to capture such dependency in natural objects by imposing more structures into the joint prior of latent variables  $s(\ell)$  and  $t(\ell)$  at all layer  $\ell$ ,  $\ell = 1, 2, \dots, L$  in the model. In particular, the joint prior  $\pi_{z|y} = p(\{z(\ell)\}_{\ell=1}^L | y)$  is given by Eqn. 1. The form of the joint prior  $\pi_{z|y}$  might look mysterious at first, but NRM parametrizes  $\pi_{z|y}$  in this particular way so that  $\pi_{z|y}$  is the conjugate prior of the model likelihood as proven in Appendix C.4. Specifically, in order to derive conjugate prior, we would like the log conditional distribution  $\log p(z(\ell) | z(\ell + 1), \dots, z(L))$  to have the linear piece-wise form as the CNNs which compute the posterior  $p(y|x)$ . This design criterion results in each term  $\langle b(t; \ell), s(\ell) \odot h(\ell) \rangle$  in the joint prior  $\pi_{z|y}$  in Eqn. 1. The conjugate form of  $\pi_{z|y}$  allows efficient inference in the NRM. Note that  $b(t; \ell)$  are the parameters of the conjugate prior  $\pi_{z|y}$ . Due to the form of  $\pi_{z|y}$ , during inference,  $b(t; \ell)$  *will become the bias terms after convolutions in CNNs* as will be shown in Theorem 3.2. Furthermore, when training in an unsupervised setup, the conjugate prior results in the RPN regularization as shown in Theorem 3.3(b). This RPN regularization helps enforce the dependencies among latent variables in the model and increases the likelihood of latent configuration presents in the data during training. For the sake of clarity, we summarize all the notations used in NRM in Table 2.

We summarize the NRM’s rendering process in Algorithm 1 below. Reconstructed images at each layer of a 5-layered NRM trained on MNIST are visualized in Figure 4. NRM reconstructs the images in two steps. First, the bottom-up E-step inference in NRM, which has a CNN form, keeps track of the optimal latent variables  $s^*(\ell)$  and  $t^*(\ell)$  from the input image. Second, in the top-down E-step reconstruction, NRM uses  $s^*(\ell)$  and  $t^*(\ell)$  to render the reconstructed image according to Eqn. 2 and 5. The network is trained using the semi-supervised learning framework discussed in Section 3.3. The reconstructed images show that NRM renders images from coarse to fine. Early layers in the model such as layer 4 and 3 capture coarse-scale features of the image while later layers such as layer 2 and 1 capture finer-scale features. Starting from layer 2, we begin to see the gist of

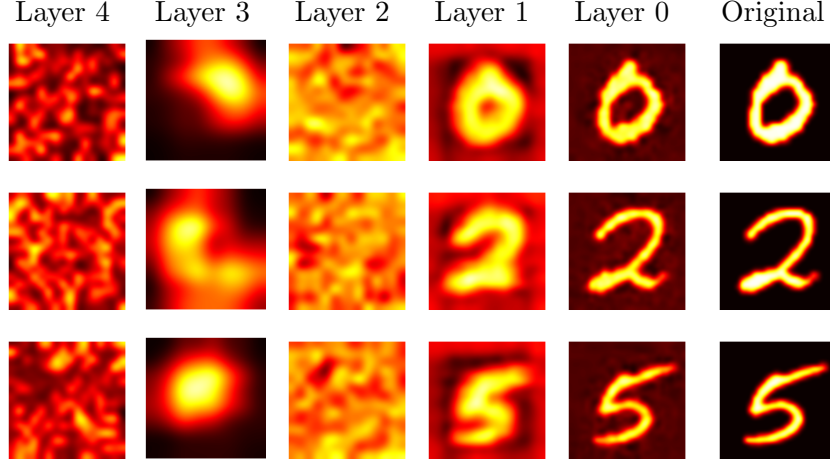


Figure 4: Reconstructed images at each layer in a 5-layer NRM trained on MNIST with 50K labeled data. Original images are in the rightmost column. Early layers in the rendering such as layer 4 and 3 capture coarse-scale features of the image while later layers such as layer 2 and 1 capture finer-scale features. From layer 2, we begin to see the gist of the rendered digits.

the rendered digits which become clearer at layer 1. Note that images at layer 4 represent the class template  $\mu_y$  in NRM, which is also the softmax weights in CNN.

In this paper, in order to simplify the notations, we only model  $p(z|y)$ . More precisely, in the subsequent sections,  $y_i$  is set to  $\arg \max_{y'} \left( \langle h(y', z_i^*; 0), x_i \rangle + \eta(y', z_i^*) + \ln \pi_{y'} \right)$  for unlabeled data.

An extension to model  $p(z, y)$  can be achieved by adding the term  $\ln \pi_y$  inside the Softmax operator in Eqn. 1. All theorems and proofs can also be easily extended to the case when  $p(z, y)$  is modeled. Furthermore, to facilitate the discussion, we will utilize  $\Gamma(\ell)$ ,  $B(\ell)$ , and  $T(t; \ell)$  to denote the set of all rendering templates  $\Gamma(\ell, p)$ , zero-padding matrices  $B(\ell, p)$ , and local translation matrices  $T(t; \ell, p)$  at layer  $\ell$ , respectively. We will also use  $\Lambda(\ell)$ ,  $\Lambda(z; \ell)$ , and  $\Lambda(t, s; \ell)$  interchangeably. Similarly, the notations  $h(y, z; \ell)$  and  $h(\ell)$ , as well as  $b(t; \ell)$  and  $b(\ell)$  are used interchangeably.

**NRM with skip connections:** In Section 3.2 below, we will show that CNNs can be derived as a joint maximum a posteriori (JMAP) inference in NRM. By introducing skip connections into the structure of the rendering matrices  $\Lambda(\ell)$ , we can also derive the convolutional neural networks with skip connections including ResNet and DenseNet. The detail derivation of ResNet and DenseNet can be found in Appendix B.7.

### 3.2 Inference

We would like to show that inference in NRM has the CNN form (see Figure 2) and, therefore, is still tractable and efficient. This correspondence between NRM and CNN helps us achieve two important goals when developing a new generative model. First, the desired generative model is complex enough with rich needed structures to capture the great diversity of forms and appearances of the objects surrounding us. Second, the inference in the model is fast and efficient so that the model can be learned in reasonable time. Such advantages justify our modeling choice of using classification models like CNNs to inform our design of generative models like NRM.

---

**Algorithm 1** Rendering Process in NRM
 

---

**Input:** Object category  $y$ .

**Output:** Rendered image  $x$  given the object category  $y$ .

**Parameters:**  $\theta = \left( \{\mu(y)\}_{y=1}^K, \{\Gamma(\ell)\}_{\ell=1}^L, \{b(\ell)\}_{\ell=1}^L \right)$  where  $\mu(y)$  is the class template,  $\Gamma(\ell)$  is the rendering template at layer  $\ell$ , and  $b(\ell)$  is the parameters of the conjugate prior  $p(z|y, x)$  at layer  $\ell$ , which turn out to be the bias terms in the ReLU after convolutions at each layer in CNNs.

1. Use Markov chain Monte Carlo method to sample the latent variables  $z(\ell)$  in NRM,  $\ell = 1, 2, \dots, L$  from  $\pi_{z|y} = \text{Softmax} \left( \frac{1}{\sigma^2} \eta(y, z) \right)$ , where  $\eta(y, z) \triangleq \sum_{\ell=1}^L \langle b(t; \ell), s(\ell) \odot h(\ell) \rangle$ .
  2. Render  $h(\ell)$ ,  $\ell = 0, 1, \dots, L - 1$  using the recursion  $h(\ell - 1) = \sum_{p \in \mathcal{P}(\ell)} s(\ell, p) T(t; \ell, p) B(\ell, p) \Gamma(\ell, p) h(\ell, p)$  in Eqn. 5, in which  $h(L) = \mu(y)$  and  $T(t; \ell, p)$  and  $B(\ell, p)$  are the local translation matrix and the zero-padding matrix at pixel location  $p$  in layer  $\ell$  as described above.
  3. Add Gaussian pixel noise  $\mathcal{N}(0, \sigma^2 \mathbf{1}_{D(0)})$  into  $h(0)$  to achieve the final rendered image  $x$ , where  $D(0)$  is the dimension, i.e. the number of pixels, of  $h(0)$  and  $x$ .
- 

The following theorem establishes the aforementioned correspondence, showing that the JMAP inference of the optimal latent variables  $z$  in NRM is indeed a CNN:

**Theorem 3.2.** Denote  $\theta := \left( \{\mu(y)\}_{y=1}^K, \{\Gamma(\ell)\}_{\ell=1}^L, \{\pi_y\}_{y=1}^K, \{b(\ell)\}_{\ell=1}^L \right)$  the set of all parameters in NRM. The JMAP inference of latent variable  $z$  in NRM is the feedforward step in CNNs. Particularly, we have:

$$\begin{aligned} \max_z \{p(z|x, y)\} &= \max_z \frac{1}{\sigma^2} \{ \langle h(y, z; 0), x \rangle + \eta(y, z) \} + \text{const} \\ &\geq \frac{1}{\sigma^2} \langle \mu(y), \psi(L) \rangle + \text{const} \end{aligned} \quad (6)$$

where  $\psi(L)$  is computed recursively. In particular,  $\psi(0) = x$  and:

$$\begin{aligned} \psi(\ell) &= \max_{z(\ell)} \left\{ \Lambda^\top(z; \ell) \psi(\ell - 1) + b(\ell) \right\} = \text{MaxPool} \left( \text{ReLU} \left( \text{Conv} \left( \Gamma^\top(\ell), \psi(\ell - 1) \right) \right) + b(\ell) \right) \\ &\stackrel{d}{=} \text{MaxPool} \left( \text{ReLU} \left( \text{Conv} (W(\ell), \psi(\ell - 1)) \right) + b(\ell) \right) \end{aligned} \quad (7)$$

The equality holds in Eqn. 6 when the parameters  $\theta$  in NRM satisfy the non-negativity assumption that the intermediate rendered image  $h(\ell) \geq 0$ ,  $\forall \ell = 1, 2, \dots, L$ .

There are four takeaways from the results in Theorem 3.2:

1. ReLU non-linearities in CNNs find the optimal value for the template selecting latent variables  $s(\ell)$  at each layer  $\ell$  in NRM, detecting if particular features exist in the image or not.
2. MaxPool operators in CNNs find the optimal value for the local translation latent variables  $t(\ell)$  at each layer  $\ell$  in NRM, locating where particular features are rendered in the image.

3. Bias terms after each convolution in CNNs are from the prior distribution of latent variables in the model. Those bias terms update the posterior estimation of latent variables from data using the knowledge encoded in the prior distribution of those latent variables.
4. Convolutions in CNNs result from reversing the local rendering operators, which use the rendering template  $\Gamma(\ell)$ , in NRM. Instead of rendering as in NRM, convolutions in CNNs perform template matching. Particularly, it can be shown that convolution weights  $W(\ell)$  in CNNs are the transposes of the rendering templates  $\Gamma(\ell)$  in NRM.

Table 1 summarizes the correspondences between NRM and CNNs. The proofs for these correspondences are postponed to Appendix C. The non-negativity assumption that the intermediate rendered image  $h(\ell) \geq 0$ ,  $\forall \ell = 1, 2, \dots, L$  allows us to apply the max-product message passing and send the max over latent variables  $z(\ell)$  operator through the product of rendering matrices  $\Lambda(z; \ell)$ . Thus, given this assumption, the equality holds in Eqn. 6. In Eqn. 7, we have removed the generative constraints inherited from NRM to derive the weights  $W(\ell)$  in CNNs, which are free parameters. As a result, when faced with training data that violates NRM’s underlying assumptions, CNNs will have more freedom to compensate. We refer to this process as a discriminative relaxation of a generative classifier [35, 36]. Finally, the dot product with the object template  $\mu(y)$  in Eqn. 6 corresponds to the fully connected layer before the Softmax non-linearity is applied in CNNs.

Given Theorem 3.2 and the four takeaways above, NRM has successfully reverse-engineered CNNs. However, the impact of Theorem 3.2 goes beyond a reverse-engineering effort. First, it provides probabilistic semantics for components in CNNs, justifying their usage, and providing an opportunity to employ probabilistic inference methods in the context of CNNs. In particular, convolution operators in CNNs can be seen as factor nodes in the factor graph associated with NRM. Similarly, activations from the convolutions in CNNs correspond to bottom-up messages in that factor graph. The bias terms added to the activations in CNNs, which are from the joint prior distribution of latent variables, are equivalent to the top-down messages from the top layers of NRM. These top-down messages have receptive fields of the whole image and are used to update the bottom-up messages, which are estimated from local information with smaller receptive fields. Finally, ReLU non-linearities and MaxPool operators in CNNs are max-marginalization operators over the template selecting and local translation latent variables  $s(\ell)$  and  $t(\ell)$  in NRM, respectively. These max-marginalization operators are from max-product message passing used to infer the latent variables in NRM.

Second, Theorem 3.2 provides a flexible framework to design CNNs. Instead of directly engineering CNNs for new tasks and datasets, we can modify NRM to incorporate our knowledge of the tasks and datasets into the model and then perform JMAP inference to achieve a new CNN architecture. For example, in Theorem 3.2, we show how ReLU can be derived from max-marginalization of  $s(\ell)$ . By changing the distribution of  $s(\ell)$ , we can derive Leaky ReLU. Furthermore, batch normalization in CNNs can be derived from NRM by normalizing intermediate rendered images at each layer in NRM. Also, as mentioned above, by introducing skip connections into the rendering matrices  $\Lambda(\ell)$ , we can derive ResNet and DenseNet. Details of those derivations can be found in Appendix B and C.

### 3.3 Learning

NRM learns from both labeled and unlabeled data. Learning in NRM can be posed as likelihood estimation problems which optimize the conditional log-likelihood  $\log p(y|x, z)$  and the expected

complete-data log-likelihood  $\mathbb{E}_z[\log p(x, z|y)]$  for supervised and unsupervised learning respectively. Interestingly, *the cross-entropy loss used in training CNNs with labeled data is the upper bound of the NRM's negative conditional log-likelihood*. NRM solves these likelihood optimization problems via the Expectation- Maximization (EM) approach. In the E-step, inference in NRM finds the optimal latent variables  $z^*$ . This inference has the form of a CNN as shown in Theorem 3.2. In the M-step, given  $z^*$ , NRM maximizes the corresponding likelihood objective functions or their lower bounds as in the case of cross-entropy loss. There is no closed-form M-step update for deep models like NRM, so NRM employs the generalized EM instead [37, 38]. In generalized EM, the M-step seeks to increase value of the likelihood objective function instead of maximizing it. In particular, in the M-step, NRM uses gradient-based methods such as Stochastic Gradient Descent (SGD) [39, 40] to update its parameters. The following theorem derives the learning objectives for NRM in both supervised and unsupervised settings.

**Theorem 3.3.** Denote  $\theta := \left( \{\mu(y)\}_{y=1}^K, \{\Gamma(l)\}_{l=1}^L, \{\pi_y\}_{y=1}^K, \{b(l)\}_{l=1}^L \right)$ . For any  $n \geq 1$ , let  $x_1, \dots, x_n$  be i.i.d. samples from NRM. Assume that the final rendered template  $h(y, z; 0)$  is normalized such that its norm is constant. The following holds:

(a) Cross-entropy loss for supervised training CNNs with labeled data:

$$\max_{(z_i)_{i=1}^n, \theta} \frac{1}{n} \sum_{i=1}^n \log p(y_i | x_i, z_i; \theta) \geq \max_{\theta} \frac{1}{n} \sum_{i=1}^n \log q(y_i | x_i) = - \min_{\theta \in \mathcal{A}_\gamma} H_{p,q}(y|x) \quad (8)$$

where  $q(y|x)$  is the posterior estimated by CNN, and  $H_{p,q}(y|x)$  is the cross-entropy between  $q(y|x)$  and the true posterior  $p(y|x)$  given by the ground truth.

(b) Reconstruction loss with RPN for unsupervised training of CNNs with labeled and unlabeled data:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\log p(x_i, z_i | y_i)] \stackrel{asympt}{\approx} \min_{\theta} \frac{1}{n} \sum_{i=1}^n \frac{\|x_i - h(y_i, z_i^*; 0)\|^2}{2} + RPN, \text{ when } \sigma \rightarrow 0 \quad (9)$$

where the latent variable  $z_i^*$  is estimated by the CNN as described in Theorem 3.2,  $h(y_i, z_i^*; 0)$  is the reconstructed image, and the RPN regularization is the negative log prior defined as follows:

$$RPN = -\frac{1}{n} \sum_{i=1}^n \log p(z_i^* | y_i) = -\frac{1}{n} \sum_{i=1}^n \text{Softmax}(\eta(y_i, z_i^*)). \quad (10)$$

### Cross-Entropy Loss for Training Convolutional Neural Networks with Labeled Data:

Part (a) of Theorem 3.3 establishes the cross-entropy loss in the context of CNNs as an upper bound of the NRM's negative conditional log-likelihood  $L_{sup} := -\frac{1}{n} \sum_{i=1}^n \log p(y_i | x_i, z_i; \theta)$ . Different from other derivations of cross-entropy loss via logistic regression, Theorem 3.3(a) derives the cross-entropy loss in conjunction with the architecture of CNNs since the estimation of the optimal latent variables  $z^*$  is part of the optimization in Eqn. 8. In other word, Theorem 3.3(a) ties feature extraction and learning for classification in CNNs into an end-to-end conditional likelihood estimation problem in NRM. This new interpretation of the cross-entropy loss suggests an interesting direction in which better losses for training CNNs with labeled data for supervised classification tasks can be derived from tighter upper bounds for  $L_{sup}$ . The Max-Min cross-entropy in Section 5 is an example. Note that the assumption that the rendered image  $h(0)$  has constant norm is solely

for the ease of presentation. Later, in Appendix B, we extend the result of Theorem 3.3(a) to the setting in which the norm of rendered image  $h(0)$  is bounded.

In order to estimate how tight the cross-entropy upper bound is, we prove the lower bound for  $L_{sup}$ . The gap between this lower bound and the cross-entropy upper bound suggests the quality of the estimation in Theorem 3.3(a). In particular, this gap is given by:

$$\frac{1}{n} \sum_{i=1}^n \left\{ \max_y \left( \max_z \left( \frac{\langle h(y, z; 0), x_i \rangle + \eta(y, z)}{\sigma^2} \right) - \frac{\langle h(y, \bar{z}; 0), x_i \rangle + \eta(y, \bar{z})}{\sigma^2} \right) \right\} + \log K, \quad (11)$$

where  $\bar{z}_i = \arg \max_{z'} p(y|x_i, z'; \theta)$  for  $1 \leq i \leq n$ . More details can be found in Appendix B while its detail proof is deferred to Appendix C.

### Reconstruction Loss with the Rendering Path Normalization (RPN) Regularization for Unsupervised Learning with Both Labeled and Unlabeled Data:

Part (b) of Theorem 3.3 suggests that NRM learns from both labeled and unlabeled data by maximizing its expected complete-data log-likelihood,  $\mathbb{E}[\log p(x_i, z_i|y_i; \theta)]$ , which is the sum of a reconstruction loss and the RPN regularization. Deriving the E-step and M-step of generalized EM when  $\sigma > 0$  is rather complicated; therefore, for the simplicity of the paper, we only focus on the setting in which  $\sigma$  goes to 0. Under that setting, in the M-step, NRM minimizes the objective function  $\frac{1}{n} \sum_{i=1}^n \frac{\|x_i - h(y_i, z_i^*; 0)\|^2}{2} + \text{RPN}$  with respect to the parameters  $\theta$  of the model. The first term in this objective function is the reconstruction loss between the input image  $x_i$  and the reconstructed template  $h(y_i, z_i^*; 0)$ . The second term is the Rendering Path Normalization (RPN) defined in Eqn. 10. RPN encourages the  $(y_i, z_i^*)$  inferred in the bottom-up E-step to have higher prior among all possible values of  $(y, z)$ . Due to the parametric form of  $p_{z|y}$  as in Eqn. 1, RPN also enforces the dependencies among latent variables  $(s(\ell), t(\ell))$  at different layers in NRM. An approximation to this RPN regularization is discussed in Appendix B.2.

### Semi-Supervised Learning with the Latent-Dependent Deep Rendering Model:

NRM learns from both labeled and unlabeled data simultaneously by maximizing a weighted combination of the cross-entropy loss for supervised learning and the reconstruction loss with RPN regularization for unsupervised learning as in Theorem 3.3. We now formulate the semi-supervised learning problem in NRM. In particular, let  $x_1, \dots, x_n$  be i.i.d. samples from NRM and assume that the labels  $y_1, \dots, y_{n_1}$  are unknown for some  $0 \leq n_1 \leq n$ , NRM utilizes the following model to determine optimal parameters employed for the semi-supervised classification task:

$$\min_{\theta} \left\{ \frac{\alpha_{\text{RC}}}{n} \sum_{i=1}^n \left( \frac{\|x_i - h(y_i, z_i^*; 0)\|^2}{2} + \text{RPN} \right) - \frac{\alpha_{\text{CE}}}{n - n_1} \sum_{i=n_1+1}^n \log q_{\theta}(y_i|x_i) \right\}, \quad (12)$$

where  $\alpha_{\text{RC}}$  and  $\alpha_{\text{CE}}$  are non-negative weights associated with the reconstruction loss/RPN regularization and the cross-entropy loss, respectively. Again, the optimal latent variables  $z_i^* = \arg \max_{z'} \left( \langle h(y_i, z'; 0), x_i \rangle + \eta(y_i, z') \right)$  are inferred in the E-step as in Theorem 3.2. For unlabeled data,  $y_i$  is set to  $\arg \max_{y'} \left( \langle h(y', z_i^*; 0), x_i \rangle + \eta(y', z_i^*) + \ln \pi_{y'} \right)$ .

In summary, combining Theorem 3.3(a) and 3.2, NRM allows us to derive CNNs with convolution layer, the ReLU non-linearity, and the MaxPool layer. These CNNs optimize the cross-entropy loss for supervised classification tasks with labeled data. Combining Theorem 3.3(b) and 3.2, NRM extends the traditional CNNs for unsupervised learning tasks in which the networks optimize the reconstruction loss with the RPN regularization. NRM does semi-supervised learning by optimizing the weighted combination of the losses in Theorem 3.3(a) and Theorem 3.3(b). Inference in the semi-supervised learning setup still follows Theorem 3.2. NRM can also be extended to explain other variants of CNNs, including ResNet and DenseNet, as well as other components in CNNs such as Leaky ReLU and batch normalization.

## 4 Statistical Guarantees for the Neural Rendering Model in the Supervised Setting

We provide statistical guarantees for NRM to establish that NRM is well defined statistically. First, we prove that NRM is consistent under a supervised learning setup. Second, we provide a generalization bound for NRM, which is proportional to the ratio of the number of active rendering paths and the total number of rendering paths in the trained NRM. A rendering path is a configuration of all latent variables in NRM as defined in Table 2, and active rendering paths are those among rendering paths  $(\hat{y}, \hat{z})$  whose corresponding rendered image is sufficiently close to one of the data point from the input data distribution. Our key results are summarized below. More details and proofs are deferred to Appendix B and C.

Governed by the connection between the cross-entropy and the posterior class probabilities  $p(y|x, z)$  under NRM, for the supervised setting of i.i.d. data  $(x_1, y_1), \dots, (x_n, y_n) \sim Q$ , where  $Q$  is the data distribution, we utilize the following model to determine optimal parameters employed for the classification task

$$\min_{\theta} \left\{ \frac{\alpha_{RC}}{n} \sum_{i=1}^n \left( \frac{\|x_i - h(y_i, z_i^*; 0)\|^2}{2} - \log p(y_i, z_i^*) \right) - \frac{\alpha_{CE}}{n} \sum_{i=1}^n \log q_{\theta}(y_i | x_i) \right\} \quad (13)$$

where  $z_i^* = \arg \max_{z'} \left( \langle h(y_i, z'; 0), x_i \rangle + \eta(y_i, z') \right)$ ,  $\alpha_{RC}$  and  $\alpha_{CE}$  are non-negative weights associated with the reconstruction loss and the cross-entropy loss respectively. Here, the approximate posterior  $q_{\theta}(y|x) = \text{Softmax} \left( \max_z \left( \langle h(y, z; 0), x \rangle + \eta(y, z) \right) + \log \pi_y \right)$  is chosen according to Theorem 3.3(a) under the regime  $\sigma \rightarrow 0$ . The optimal solutions of objective function (13) induce a corresponding set of optimal (active) rendering paths that play a central role for an understanding of generalization bound regarding the classification tasks.

Before proceeding to the generalization bound, we first state the informal result regarding the consistency of optimal solutions of (13) when the sample size  $n$  goes to infinity.

**Theorem 4.1.** *(Informal) Under the appropriate conditions regarding parameter spaces of  $\theta$ , the optimal solutions of objective function (13) converge almost surely to those of the following population objective function*

$$\min_{\theta} \left\{ \alpha_{RC} \left( \int \left( \frac{\|x - h(y, z^*; 0)\|^2}{2} - \log p(y, z^*) \right) dQ(x, y) \right) - \alpha_{CE} \int \log q_{\theta}(y|x) dQ(x, y) \right\}$$



where  $z^* = \arg \max_{z'} \left( \langle h(y, z'; 0), x \rangle + \eta(y, z') \right)$ .

In Appendix C, we provide detail formulations of Theorem 4.1 for the supervised learning. Additionally, the detail proof of this result is presented in Appendix D. The statistical guarantee regarding optimal solutions of (13) validates their usage for the classification task. Given that NRM is consistent under a supervised learning setup, the following theorem establish a generalization bound for the model.

**Theorem 4.2.** (Informal) Let  $L_{\mathcal{A}}$  and  $L_{\mathcal{D}}$  denote the population and empirical losses on the data population  $\mathcal{A}$  and the training set  $\mathcal{D}$  of NRM, respectively. Under the margin-based loss, the generalization gap of the classification framework with optimal solutions from (13) is controlled by the following term

$$L_{\mathcal{A}} \lesssim L_{\mathcal{D}} + \frac{8K(2K-1)}{\sqrt{n}} \left( 2\bar{\tau}_n |\mathcal{L}| (R^2 + 1) + |\log \bar{\gamma}| \right) + \sqrt{\frac{\log(2\delta^{-1})}{2n}}$$

with probability  $1 - \delta$ . Here,  $\bar{\tau}_n \in (0, 1)$  denotes the ratio of active optimal rendering paths among all the optimal rendering paths,  $|\mathcal{L}|$  is the total number of rendering paths,  $\bar{\gamma}$  is the lower bound of prior probability  $\pi(y)$  regarding labels, and  $R$  is the radius of the sphere that the rendered images belong to.

The detail formulations of the above theorem are postponed to Appendix C. The dependence of generalization bound on the number of active rendering paths  $\bar{\tau}_n |\mathcal{L}|$  helps to justify our modeling assumptions. In particular, NRM helps to reduce the number of active rendering paths thanks to the dependencies among its latent variables, thereby tightening the generalization bound. Nevertheless, there is a limitation regarding the current generalization bound. In particular, the bound involves the number of rendering paths  $|\mathcal{L}|$ , which is usually large. This is mainly because our bound has not fully taken into account the structures of CNNs, which is the limitation shared among other latest generalization bounds for CNN. It is interesting to explore if techniques in works by [41] and [42] can be employed to improve the term  $|\mathcal{L}|$  in our bound.

**Extension to unsupervised and semi-supervised settings** Apart from the statistical guarantee and generalization bound established for the supervised setting, we also provide careful theoretical studies as well as detailed proofs regarding these results for the unsupervised and semi-supervised setting in Appendix B, Appendix C, Appendix D, and Appendix E.

## 5 New Max-Min Cross Entropy From The Neural Rendering Model

In this section, we explore a particular way to derive an alternative to cross-entropy inspired by the results in Theorem 3.3(a). In particular, denoting  $z^{\max} \triangleq \arg \max_z \{ \langle h(y, z; 0), x \rangle + \eta(y, z) \}$  and  $z^{\min} \triangleq \arg \min_z \{ \langle h(y, z; 0), x \rangle + \eta(y, z) \}$ , the new cross-entropy  $H^{\text{M\&M}}$ , which is called the *Max-Min cross-entropy*, is the weighted average of the cross-entropy losses from  $z^{\max}$  and  $z^{\min}$ :

$$H^{\text{M\&M}} \triangleq \alpha^{\max} H_{p,q}(y|x, z^{\max}) + \alpha^{\min} H_{p,q}(y|x, z^{\min}) = \alpha^{\max} H_{p,q}^{\max}(y|x) + \alpha^{\min} H_{p,q}^{\min}(y|x).$$

Here the Max cross-entropy  $H_{p,q}^{\max}$  and Min cross entropy  $H_{p,q}^{\min}$  maximizes the correct target posterior and minimizes the incorrect target posterior, respectively. Similar to the cross-entropy loss, the

Max-Min cross-entropy can also be shown to be an upper bound to the negative conditional log-likelihood  $L_{\text{sup}}$  of the NRM and has the same generalization bound derived in Section 4. The Max-Min networks in Figure 5 realize this new loss. These networks have two CNN-like branches that share weights. The max branch estimates  $z^{\text{max}}$  using ReLU and Max-Pooling, and the min branch estimates  $z^{\text{min}}$  using the Negative ReLU, i.e.,  $\min(\cdot, 0)$ , and Min-Pooling. The Max-Min networks can be interpreted as a form of knowledge distillation like the Born Again networks [43] and the Mean Teacher networks. However, instead of a student network learning from a teacher network, in Max-Min networks, two students networks, the Max and the Min networks, cooperate and learn from each other during the training.

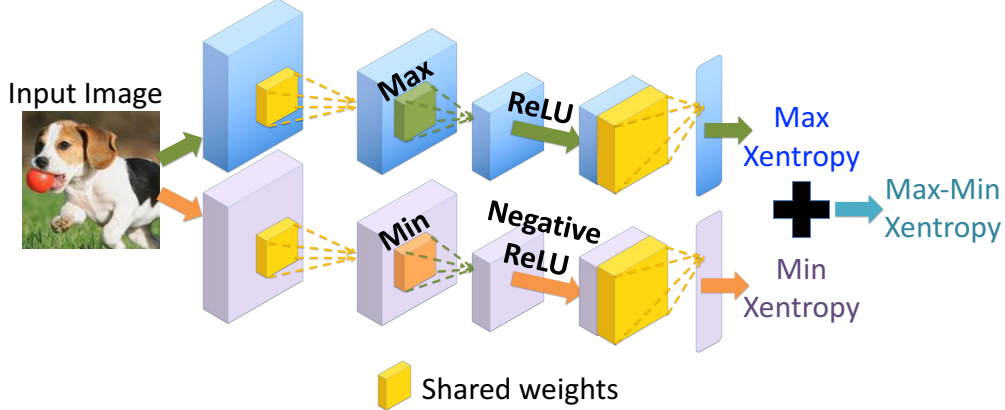


Figure 5: The Max-Min network. The Max branch in the Max-Min network maximizes the correct target posterior while the Min branch minimizes the incorrect target posterior. These two networks share weights. The Max-Min cross-entropy loss is the weighted average of the cross-entropy losses from the Max and the Min networks.

## 6 Experiments

### 6.1 Semi-Supervised Learning

We show NRM armed with Max-Min cross-entropy and Mean Teacher regularizer achieves SOTA on benchmark datasets. We discuss the experimental results for CIFAR10 and CIFAR100 here. The results for SVHN, training losses, and training details, can be found in the Appendix A & F.

**CIFAR-10:** Table 3 shows comparable results of NRM to SOTA methods. NRM is also better than the best methods that do not use consistency regularization like GAN, Ladder network, and ALI when using only  $N_l=2K$  and  $4K$  labeled images. NRM outperform DRM in all settings. Also, among methods in our comparison, NRM achieves the best test accuracy when using all available labeled data ( $N_l=50K$ ). We hypothesize that NRM has the advantage over consistency regularization methods like Temporal Ensembling and Mean Teacher when there are enough labeled data is because the consistency regularization in those methods tries to match the activations in the network, but does not take into account the available class labels. On the contrary, NRM employs the class labels, if they are available, in its reconstruction loss and RPN regularization as in Eqns. 9 and 10. In all settings, RPN regularizer improves NRM performance. Even though

Table 3: Error rate percentage on CIFAR-10 over 3 runs.

	1K labels 50K images	2K labels 50K images	4K labels 50K images	50K labels 50K images
Adversarial Learned Inference [8]	$19.98 \pm 0.89$	$19.09 \pm 0.44$	$17.99 \pm 1.62$	
Improved GAN [7]	$21.83 \pm 2.01$	$19.61 \pm 2.09$	$18.63 \pm 2.32$	
Ladder Network [44]			$20.40 \pm 0.47$	
$\Pi$ model [5]	$27.36 \pm 1.20$	$18.02 \pm 0.60$	$13.20 \pm 0.27$	$6.06 \pm 0.11$
Temporal Ensembling [5]			$12.16 \pm 0.31$	$5.60 \pm 0.10$
Mean Teacher [6]	$21.55 \pm 1.48$	$15.73 \pm 0.31$	$12.31 \pm 0.28$	$5.94 \pm 0.15$
VAT+EntMin [45]			<b>10.55</b>	
DRM [15, 46]	$27.67 \pm 1.86$	$20.71 \pm 0.30$	$15.36 \pm 0.34$	$5.75 \pm 0.24$
Supervised-only	$46.43 \pm 1.21$	$33.94 \pm 0.73$	$20.66 \pm 0.57$	$5.82 \pm 0.15$
NRM without RPN	$24.88 \pm 0.76$	$18.97 \pm 0.80$	$14.41 \pm 0.19$	$5.57 \pm 0.07$
NRM+RPN	$24.48 \pm 0.43$	$18.62 \pm 0.70$	$14.18 \pm 0.46$	$5.35 \pm 0.08$
NRM+RPN+Max-Min	$21.55 \pm 0.46$	$16.24 \pm 0.17$	$12.50 \pm 0.35$	<b><math>4.85 \pm 0.10</math></b>
NRM+RPN+Max-Min+Mean Teacher	<b><math>19.79 \pm 0.74</math></b>	<b><math>15.11 \pm 0.51</math></b>	$11.81 \pm 0.13$	$4.88 \pm 0.09$

the improvement from RPN is small, it is consistent across the experiments. Furthermore, using Max-Min cross-entropy significantly reduces the test errors. When combining with Mean-Teacher, our Max-Min NRM improves upon Mean-Teacher and consistently achieves either SOTA results or second best results in all settings. This consistency in performance is only observed in our method and Mean-Teacher. Also, like with Mean-Teacher, NRM can potentially be combined with other consistency regularization methods, e.g., the Virtual Adversarial Training (VAT) [45], to obtain better results.

**CIFAR-100:** Table 4 shows NRM’s comparable results to  $\Pi$  model and Temporal Ensembling, as well as better results than DRM. Same as with CIFAR10, using the RPN regularizer results in a slightly better test accuracy, and NRM achieves better results than  $\Pi$  model and Temporal Ensembling method when using all available labeled data. Notice that combining with Mean-Teacher just slightly improves NRM’s performance when training with 10K labeled data. This is again because consistency regularization methods like Mean-Teacher do not add much advantage when there are enough labeled data. However, NRM+Max-Min still yields better test errors and achieves SOTA result in all settings. Note that since combining with Mean-Teacher does not help much here, we only show result for NRM+Max-Min.

## 6.2 Supervised Learning with Max-Min Cross-Entropy

The Max-Min cross-entropy can be applied not only to improve semi-supervised learning but also on deep models including CNNs to enhance their supervised learning performance. In our experiments, we indeed observe Max-Min cross-entropy reduces the test error for supervised object classification on CIFAR10. In particular, using the Max-Min cross-entropy loss on a 29-layer ResNet [47] trained with the Shake-Shake regularization [48] and Cutout data augmentation [49], we are able to achieve SOTA test error of 2.30% on CIFAR10, an improvement of 0.26% over the test error of the baseline architecture trained with the traditional cross-entropy loss. While 0.26% improvement seems small, it is a meaningful enhancement given that our baseline architecture (ResNeXt + Shake-Shake + Cutout) is the second best model for supervised learning on CIFAR10. Such small improvement over

Table 4: Error rate percentage on CIFAR-100 over 3 runs.

	10K labels 50K images	50K labels 50K images
$\Pi$ model [5]	$39.19 \pm 0.36$	$26.32 \pm 0.04$
Temporal Ensembling [5]	$38.65 \pm 0.51$	$26.30 \pm 0.15$
DRM [15, 46]	$41.09 \pm 0.31$	$27.06 \pm 0.19$
Supervised-only	$44.56 \pm 0.30$	$26.42 \pm 0.17$
NRM without RPN	$40.70 \pm 1.13$	$26.27 \pm 0.09$
NRM+RPN	$39.85 \pm 0.46$	$25.84 \pm 0.10$
NRM+RPN+Mean Teacher	$39.84 \pm 0.32$	$25.98 \pm 0.35$
NRM+RPN+Max-Min	<b><math>37.75 \pm 0.66</math></b>	<b><math>24.38 \pm 0.29</math></b>

an already very accurate model is significant in applications in which high accuracy is demanded such as self-driving cars or medical diagnostics. Similarly, we observe Max-Min improves the top-5 test error of the Squeeze-and-Excitation ResNeXt-50 network [50] on ImageNet by 0.17% compared to the baseline (7.04% vs. 7.21%). For a fair comparison, we re-train the baseline models and report the scores in the re-implementation.

## 7 Discussion

We present the NRM, a general and an effective framework for semi-supervised learning that combines generation and prediction in an end-to-end optimization. Using NRM, we can explain operations used in CNNs and develop new features that help learning in CNNs. For example, we derive the new Max-Min cross-entropy loss for training CNNs, which outperforms the traditional cross-entropy.

In addition to the results discussed above, there are still many open problems related to NRM that have not been addressed in the paper. We give several examples below:

- An adversarial loss like in GANs can be incorporated into the NRM so that the model can generate realistic images. Furthermore, more knowledge of image generation from graphics and physics can be integrated in NRM so that the model can employ more structures to help learning and generation.
- The unsupervised and (semi)-supervised models that we consider throughout the paper are under the assumption that the noise  $\sigma$  of NRM goes to 0. Governed by this assumption, we are able to derive efficient inference algorithms as well as rigorous statistical guarantees with these models. For the setting that  $\sigma$  is not close to 0, the inference with these models will rely on vanilla Expectation-Maximization (EM) algorithm for mixture models to obtain reliable estimators for the rendering templates. Since the parameters of interest are being shared among different rendering templates and have high dimensional structures, it is of practical interest to develop efficient EM algorithm to capture these properties of parameters under that setting of  $\sigma$  in NRM.
- Thus far, the statistical guarantees with parameter estimation in the paper are established under the ideal assumptions that the optimal global solutions are obtained. However, it happens in practice that the inference algorithms based on SGD with the unsupervised and

(semi)-supervised models usually lead to (bad) local minima. As a consequence, investigating sufficient conditions for the inference algorithms to avoid being trapped at bad local minima is an important venue of future work.

- NRM hinges upon the assumption that the data is generated from mixture of Gaussian distributions with the mean parameters characterizing the complex rendering templates. However, it may happen in reality that the underlying distribution of each component of mixtures is not Gaussian distribution. Therefore, extending the current understandings with NRM under Gaussian distribution to other choices of underlying distributions is an interesting direction to explore.

We would like to end the paper with a remark that NRM is a flexible framework that enables us to introduce new components in the generative process and the corresponding features for CNNs can be derived in the inference. This hallmark of NRM provide a more fundamental and systematic way to design and study CNNs.

## 8 Acknowledgements

First of all, we are very grateful to Amazon AI for providing a highly stimulating research environment for us to start this research project and further supporting our research through their cloud credits program. We would also like to express our sincere thanks to Gautam Dasarathy for great discussions. Furthermore, we would also like to thank Doris Y. Tsao for suggesting and providing references for connections between our model and feedforward and feedback connections in the brain.

Many people during Tan Nguyen’s internship at Amazon AI have helped by providing comments and suggestions on our work, including Stefano Soatto, Zack C. Lipton, Yu-Xiang Wang, Kamyar Azizzadenesheli, Fanny Yang, Jean Kossaifi, Michael Tschannen, Ashish Khetan, and Jeremy Bernstein. We also wish to thank Sheng Zha who has provided immense help with MXNet framework to implement our models.

Finally, we would like to thank members of DSP group at Rice, Machine Learning group at UC Berkeley, and Anima Anandkumar’s TensorLab at Caltech who have always been supportive throughout the time it has taken to finish this project.

## References

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, pp. 2672–2680, 2014.
- [2] R. P. N. Rao and D. H. Ballard, “Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects,” *Nature Neuroscience*, vol. 2, pp. 79 EP –, 01 1999.
- [3] K. Friston, “Does predictive coding have a future?,” *Nature Neuroscience*, vol. 21, no. 8, pp. 1019–1021, 2018.
- [4] B. Neyshabur, R. R. Salakhutdinov, and N. Srebro, “Path-sgd: Path-normalized optimization in deep neural networks,” in *Advances in Neural Information Processing Systems*, pp. 2422–2430, 2015.
- [5] S. Laine and T. Aila, “Temporal ensembling for semi-supervised learning,” in *International Conference on Learning Representations*, 2017.
- [6] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Advances in Neural Information Processing Systems 30*, pp. 1195–1204, 2017.
- [7] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, “Improved techniques for training gans,” in *Advances in Neural Information Processing Systems 29*, pp. 2234–2242, 2016.
- [8] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, “Adversarially learned inference,” in *International Conference on Learning Representations*, 2017.
- [9] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [10] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” in *Advances in Neural Information Processing Systems 27*, pp. 3581–3589, 2014.
- [11] J. Donahue, P. Krähenbühl, and T. Darrell, “Adversarial feature learning,” in *International Conference on Learning Representations*, 2017.
- [12] L. Dinh, D. Krueger, and Y. Bengio, “Nice: Non-linear independent components estimation,” in *International Conference on Learning Representations Workshop*, 2015.
- [13] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real nvp,” in *International Conference on Learning Representations*, 2017.
- [14] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1x1 convolutions,” in *Advances in Neural Information Processing Systems*, 2018.

- [15] A. B. Patel, M. T. Nguyen, and R. Baraniuk, “A probabilistic framework for deep learning,” in *Advances in Neural Information Processing Systems 29*, pp. 2558–2566, 2016.
- [16] A. Achille and S. Soatto, “Emergence of invariance and disentanglement in deep representations,” *The Journal of Machine Learning Research*, vol. 19, pp. 1–34, 2018.
- [17] A. Achille and S. Soatto, “Information dropout: Learning optimal representations through noisy computation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [18] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” in *Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377, 1999.
- [19] V. Pappas, Y. Romano, and M. Elad, “Convolutional neural networks analyzed via convolutional sparse coding,” *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 2887–2938, 2017.
- [20] H. Bristow, A. Eriksson, and S. Lucey, “Fast convolutional sparse coding,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 391–398, IEEE, 2013.
- [21] B. Wohlberg, “Efficient convolutional sparse coding,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7173–7177, IEEE, 2014.
- [22] F. Heide, W. Heidrich, and G. Wetzstein, “Fast and flexible convolutional sparse coding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5135–5143, 2015.
- [23] V. Pappas, J. Sulam, and M. Elad, “Working locally thinking globally: Theoretical guarantees for convolutional sparse coding,” *IEEE Transactions on Signal Processing*, vol. 65, no. 21, pp. 5687–5701, 2017.
- [24] J. Bruna and S. Mallat, “Invariant scattering convolution networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.
- [25] S. Mallat, “Understanding deep convolutional networks,” *Phil. Trans. R. Soc. A*, vol. 374, no. 2065, p. 20150203, 2016.
- [26] S. Arora, N. Cohen, and E. Hazan, “On the optimization of deep networks: Implicit acceleration by overparameterization,” *arXiv preprint arXiv:1802.06509*, 2018.
- [27] C. D. Freeman and J. Bruna, “Topology and geometry of half-rectified network optimization,” in *International Conference on Learning Representations*, 2017.
- [28] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, “The loss surfaces of multilayer networks,” in *Artificial Intelligence and Statistics*, pp. 192–204, 2015.
- [29] K. Kawaguchi, “Deep learning without poor local minima,” in *Advances in Neural Information Processing Systems 29*, pp. 586–594, 2016.
- [30] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, “Generalization in deep learning,” *arXiv preprint arXiv:1710.05468*, 2017.
- [31] B. Neyshabur, S. Bhojanapalli, D. Mcallester, and N. Srebro, “Exploring generalization in deep learning,” in *Advances in Neural Information Processing Systems 30*, pp. 5947–5956, 2017.

- [32] R. Balestrierio and R. G. Baraniuk, “A spline theory of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning*, 2018.
- [33] R. Vidal, J. Bruna, R. Giryes, and S. Soatto, “Mathematics of deep learning,” *arXiv preprint arXiv:1712.04741*, 2017.
- [34] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *International Conference on Machine Learning*, pp. 1050–1059, 2016.
- [35] A. Y. Ng and M. I. Jordan, “On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes,” in *Advances in Neural Information Processing Systems*, pp. 841–848, 2002.
- [36] J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, “Generative or discriminative? getting the best of both worlds,” *Bayesian statistics*, vol. 8, no. 3, pp. 3–24, 2007.
- [37] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 39, pp. 1–38, 1997.
- [38] C. M. Bishop, *Pattern recognition and machine learning (information science and statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.
- [39] H. Robbins and S. Monro, “A stochastic approximation method,” in *Herbert Robbins Selected Papers*, pp. 102–109, Springer, 1985.
- [40] J. Kiefer, J. Wolfowitz, *et al.*, “Stochastic estimation of the maximum of a regression function,” *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.
- [41] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, “Spectrally-normalized margin bounds for neural networks,” *Advances in Neural Information Processing Systems (NIPS)*, 2017.
- [42] N. Golowich, A. Rakhlin, and O. Shamir, “Size-independent sample complexity of neural networks,” *Proceedings of the Conference On Learning Theory (COLT)*, 2018.
- [43] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, “Born-again neural networks,” *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.
- [44] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, “Semi-supervised learning with ladder networks,” in *Advances in Neural Information Processing Systems 28*, pp. 3546–3554, 2015.
- [45] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama, “Virtual adversarial training: a regularization method for supervised and semi-supervised learning,” *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [46] A. B. Patel, T. Nguyen, and R. G. Baraniuk, “A probabilistic theory of deep learning,” *arXiv preprint arXiv:1504.00641*, 2015.



- [47] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5987–5995, IEEE, 2017.
- [48] X. Gastaldi, “Shake-shake regularization of 3-branch residual networks,” in *International Conference on Learning Representations*, 2017.
- [49] T. DeVries and G. W. Taylor, “Improved regularization of convolutional neural networks with cutout,” *arXiv preprint arXiv:1708.04552*, 2017.
- [50] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [51] A. Kumar, P. Sattigeri, and T. Fletcher, “Semi-supervised learning with gans: Manifold invariance with improved inference,” in *Advances in Neural Information Processing Systems 30*, pp. 5534–5544, 2017.
- [52] T. Nguyen, W. Liu, E. Perez, R. G. Baraniuk, and A. B. Patel, “Semi-supervised learning with the deep rendering mixture model,” *arXiv preprint arXiv:1612.01942*, 2016.
- [53] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational inference: A review for statisticians,” *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.
- [54] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [55] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [56] S. van de Geer, *Empirical Processes in M-estimation*. Cambridge University Press, 2000.
- [57] R. Vershynin, “Introduction to the non-asymptotic analysis of random matrices,” *arXiv:1011.3027v7*, 2011.
- [58] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. Stochastic Modelling and Applied Probability, Springer, 1996.
- [59] R. Dudley, “Central limit theorems for empirical measures,” *Annals of Probability*, vol. 6, 1978.
- [60] V. Koltchinskii and D. Panchenko, “Empirical margin distributions and bounding the generalization error of combined classifiers,” *Annals of Statistics*, vol. 30, 2002.
- [61] A. W. van der Vaart and J. Wellner, *Weak Convergence and Empirical Processes*. New York, NY: Springer-Verlag, 1996.
- [62] I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts,” in *International Conference on Learning Representations*, 2016.

- [63] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

## Supplementary Material

### A Appendix A

This appendix contains semi-supervised learning results of NRM on SVHN compared to other methods.

#### Semi-Supervised Learning Results on SVHN:

Table 5: Error rate percentage on SVHN in comparison with other state-of-the-art methods. All results are averaged over 2 runs (except for NRM+RPN when using all labels, 1 run)

	250 labels 73257 images	500 labels 73257 images	1000 labels 73257 images	73257 labels 73257 images
ALI [8]			$7.42 \pm 0.65$	
Improved GAN [7]		$18.44 \pm 4.8$	$8.11 \pm 1.3$	
+ Jacob.-reg + Tangents [51]		$4.87 \pm 1.60$	$4.39 \pm 1.20$	
$\Pi$ model [5]	$9.69 \pm 0.92$	$6.83 \pm 0.66$	$4.95 \pm 0.26$	$2.50 \pm 0.07$
Temporal Ensembling [5]		$5.12 \pm 0.13$	$4.42 \pm 0.16$	$2.74 \pm 0.06$
Mean Teacher [6]	$4.35 \pm 0.50$	$4.18 \pm 0.27$	$3.95 \pm 0.19$	<b><math>2.50 \pm 0.05</math></b>
VAT+EntMin [45]			3.86	
DRM [52]		9.85	6.78	
Supervised-only	$27.77 \pm 3.18$	$16.88 \pm 1.30$	$12.32 \pm 0.95$	$2.75 \pm 0.10$
NRM without RPN	$9.78 \pm 0.24$	$7.42 \pm 0.61$	$5.64 \pm 0.13$	$3.46 \pm 0.04$
NRM+RPN	$9.28 \pm 0.01$	$6.56 \pm 0.88$	$5.47 \pm 0.14$	3.57
NRM+RPN+Max-Min+MeanTeacher	<b><math>3.97 \pm 0.21</math></b>	<b><math>3.84 \pm 0.34</math></b>	<b><math>3.70 \pm 0.04</math></b>	$2.87 \pm 0.05$

## B Appendix B

In this appendix, we give further connection of NRM to cross entropy as well as additional derivation of NRM to various models under both unsupervised and (semi)-supervised setting of data being mentioned in the main text. We also formally present our results on consistency and generalization bounds for NRM in supervised and semi-supervised learning settings. In addition, we explain how to extend NRM to derive ResNet and DenseNet. For the simplicity of the presentation, we denote  $\theta = \left( \{\mu(y)\}_{y=1}^K, \{\Gamma(\ell)\}_{\ell=1}^L, \{\pi_y\}_{y=1}^K, \{b(\ell)\}_{\ell=1}^L \right)$  to represent all the parameters that we would like to estimate from NRM where  $\mathcal{L}$  is the set of all possible values of latent (nuisance) variables  $z = (t(\ell), s(\ell))_{\ell=1}^L$ . Additionally, for each  $(y, z) \in \mathcal{J} := \{1, \dots, K\} \times \mathcal{L}$ , we denote  $\theta_{y,z} = \left( \mu(y), \{\Gamma(\ell)\}_{\ell=1}^L, \{b(\ell)\} \right)$ , i.e., the subset of parameters corresponding to specific label  $y$  and latent variable  $z$ . Furthermore, to stress the dependence of  $\eta(y, z)$  on  $\theta$ , we define the following function

$$\tau(\theta_{y,z}) := \eta(y, z) = \sum_{\ell=1}^L b^\top(\ell) (s(\ell) \odot z(\ell)) = \sum_{\ell=1}^L b^\top(\ell) M(s; \ell) z(\ell)$$

for each  $(y, z) \in \mathcal{J}$  where  $M(s; \ell) = \text{diag}(s(\ell))$  is a masking matrix associated with  $s(\ell)$ . Throughout this supplement, we will use  $\tau(\theta_{y,z})$  and  $\eta(y, z)$  interchangeably as long as the context is clear. Furthermore, we assume that  $\Gamma(\ell) \in \theta_\ell$ , which is a subset of  $\mathbb{R}^{F(\ell) \times D(\ell)}$  for any  $1 \leq \ell \leq L$ ,  $\mu(y) \in \Omega$ , which is a subset of  $\mathbb{R}^{D(\ell)}$ , for  $1 \leq y \leq K$ , and  $b(t; \ell) \in \Xi(\ell)$ , which is a subset of  $\mathbb{R}^{D(\ell)}$  for all choices of  $t(\ell)$  and  $1 \leq \ell \leq L$ . Last but not least, we say that  $\theta$  satisfies the *non-negativity assumption* if the intermediate rendered images  $z(\ell)$  satisfy  $z(\ell) \geq 0$  for all  $1 \leq \ell \leq L$ . Finally, we use  $A^\top$  to denote transpose of the matrix  $A$ .

### B.1 Connection between NRM and cross entropy

As being established in part (a) of Theorem 3.3, the cross entropy is the lower bound of maximizing the conditional log likelihood. In the following full theorem, we will show both the upper bound and the lower bound of maximizing the conditional log likelihood in terms of the cross entropy.

**Theorem B.1.** *Given any  $\gamma > 0$ , we denote  $\mathcal{A}_\gamma = \{\theta : \|h(y, z; 0)\| = \gamma\}$ . For any  $n \geq 1$  and  $\sigma > 0$ , let  $x_1, \dots, x_n$  be i.i.d. samples from the NRM. Then, the following holds*

(a) (Lower bound)

$$\begin{aligned} & \max_{(z_i)_{i=1}^n, \theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log p(y_i | x_i, z_i; \theta) \\ & \geq \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log \left( \text{Softmax} \left( \max_{z_i} \left( \frac{h^\top(y_i, z_i; 0) x_i + \eta(y_i, z_i)}{\sigma^2} \right) + b_{y_i} \right) \right) \\ & = \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log q(y_i | x_i) = - \min_{\theta \in \mathcal{A}_\gamma} H_{p,q}(y|x) \end{aligned}$$

where  $b_y = \log \pi_y$  for all  $1 \leq y \leq K$ ,  $q(y|x) = \text{Softmax} \left( \max_z \left( h^\top(y, z; 0) x + \eta(y, z) \right) / \sigma^2 + b_y \right)$  for all  $(x, y)$ , and  $H_{p,q}(y|x)$  is the cross-entropy between the estimated posterior  $q(y|x)$  and the true

posterior given by the ground-true labels  $p(y|x)$ .

(b) (Upper bound)

$$\begin{aligned} \max_{(z_i)_{i=1}^n, \theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log p(y_i|x_i, z_i; \theta) \\ \leq \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \left\{ \log q(y_i|x_i) + \max_y \left( \max_{z_i} \left( \frac{h^\top(y, z_i; 0)x_i + \eta(y, z_i)}{\sigma^2} \right) \right. \right. \\ \left. \left. - \frac{h^\top(y, \bar{z}_i; 0)x_i + \eta(y, \bar{z}_i)}{\sigma^2} \right) \right\} + \log K \end{aligned}$$

where  $\bar{z}_i = \arg \max_{z_i} p(y_i|x_i, z_i; \theta)$  for  $1 \leq i \leq n$ .

*Remark B.2.* As being demonstrated in Theorem 3.2,  $\max_z (h^\top(y, z; 0)x + \eta(y, z)) / \sigma^2$ , approximately, has the form of a CNN. If we further have the non-negativity assumption with  $\theta$ , then this is exact. Therefore, the cross entropy  $H_{p,q}$  obtained in Theorem B.1 has a strong connection with CNN.

*Remark B.3.* The gap between the upper bound and the lower bound of maximizing the conditional log likelihood in terms of cross entropy function suggests how good the estimation in Theorem B.1 is. In particular, this gap is given by:

$$\frac{1}{n} \sum_{i=1}^n \left\{ \underbrace{\max_y \left( \max_{z_i} \left( \frac{h^\top(y, z_i; 0)x_i + \eta(y, z_i)}{\sigma^2} \right) - \frac{h^\top(y, \bar{z}_i; 0)x_i + \eta(y, \bar{z}_i)}{\sigma^2} \right)}_{\text{trade-off loss}} \right\} + \log K,$$

where  $\bar{z}_i = \arg \max_{z_i} p(y_i|x_i, z_i; \theta)$  for  $1 \leq i \leq n$ . As long as the number of labels is not too large and the trade-off loss is sufficiently small, the gap between the upper bound and the lower bound in Theorem B.1 is small.

## B.2 Learning in the NRM without noise for unsupervised setting under non-negativity assumption

To ease the presentation of the inference with NRM without noise, we first assume that rendered images  $h(y, z; 0)$  satisfy the non-negativity assumption (Later in Section B.3, we will discuss the relaxation of this assumption for the inference of the NRM). With this assumption, as being demonstrated in Theorem 3.2, we have:

$$\max_{y, z} \left\{ h^\top(y, z; 0)x + \eta(y, z) \right\} = \max_y h^\top(y) \psi(L) \quad (14)$$

where we define

$$\begin{aligned} \psi(L) = \max_{z(L)} \Lambda^\top(z(L)) \left( \max_{z(L-1)} \left( \Lambda^\top(z; L-1) \cdots \left( \max_{z(1)} \Lambda^\top(z; 1)x + b(1) \right) \cdots \right) + b(L-1) \right) \\ + b(L). \end{aligned}$$

Now, we will provide careful derivation of part (b) of Theorem 3.3 in the main text. Remind that, for the unsupervised setting, we have data  $x_1, \dots, x_n$  are i.i.d. samples from NRM. The complete-data log-likelihood of the NRM is given as follows:

$$\mathbb{E}_{y,z} [\log p(x, (y, z))] = \sum_{i=1}^n \sum_{(y,z) \in \mathcal{J}} P(y, z | x_i) \{ \log \pi_{y,z} + \log \mathcal{N}(x_i | h(y, z; 0)) \}$$

where we have

$$\begin{aligned} P(y, z | x_i) &= \frac{\pi_{y,z} \mathcal{N}(x_i | h(y, z; 0))}{\sum_{(y', z') \in \mathcal{J}} \pi_{y', z'} \mathcal{N}(x_i | h(y', z'; 0))} \\ &= \frac{\pi_y \exp\left(-\frac{\|x_i - h(y, z; 0)\|^2 - 2\eta(y, z)}{2\sigma^2}\right)}{\sum_{(y', z') \in \mathcal{J}} \pi_{y'} \exp\left(-\frac{\|x_i - h(y', z'; 0)\|^2 - 2\eta(y', z')}{2\sigma^2}\right)}. \end{aligned}$$

At the zero-noise limit, i.e.,  $\sigma \rightarrow 0$ , it is clear that  $P(y, z | x_i) = 1$  as  $(y, z) = \arg \min_{(y', z') \in \mathcal{J}} \left\{ \|x_i - h(y', z'; 0)\|^2 - 2\eta(y', z') \right\}$  and  $P(y, z | x_i) = 0$  otherwise. Therefore, we can asymptotically view the complete log-likelihood of the NRM under the zero-noise limit as

$$\begin{aligned} &\sum_{i=1}^n \sum_{(y,z) \in \mathcal{J}} r_{y,z} \left( \log \pi_{y,z} - \frac{1}{2} \|x_i - h(y, z; 0)\|^2 \right) \\ &= \underbrace{\sum_{i=1}^n \sum_{(y,z) \in \mathcal{J}} -\frac{1}{2} r_{y,z} \|x_i - h(y, z; 0)\|^2}_{\text{Reconstruction Loss}} + \underbrace{\sum_{i=1}^n \sum_{(y,z) \in \mathcal{J}} r_{y,z} \log \pi_{y,z}}_{\text{Path Normalization Regularizer}} \end{aligned}$$

where

$$r_{y,z} \equiv \begin{cases} 1, & \text{if } (y, z) = \arg \min_{(y', z') \in \mathcal{J}} \left\{ \|x_i - h(y', z'; 0)\|^2 - 2\eta(y', z') \right\} \\ 0, & \text{otherwise} \end{cases}$$

With the above formulation, we have the following objective function

$$U_n = \min_{\theta} \frac{1}{n} \sum_{i=1}^n \sum_{(y,z) \in \mathcal{J}} r_{y,z} \left( \frac{1}{2} \|x_i - h(y, z; 0)\|^2 - \log \pi_{y,z} \right) \quad (15)$$

where  $\theta = \left( \{\mu(y)\}_{y=1}^K, \{\Gamma(\ell)\}_{\ell=1}^L, \{\pi_y\}_{y=1}^K, \{b(\ell)\}_{\ell=1}^L \right)$ . We call the above objective function to be *unsupervised NRM without noise*.

**Relaxation of unsupervised NRM without noise:** Unfortunately, the inference with unsupervised NRM without noise is intractable in practice due to two elements: the involvement of  $\|h(y', z'; 0)\|^2$  to determine the value of  $r_{y,z}$  and the summation  $\sum_{(y', z') \in \mathcal{J}} \exp(\eta(y', z') + \log \pi_{y'})$  in the denominator of  $\pi_{y,z}$  for all  $(y, z) \in \mathcal{J}$ . Therefore, we need to develop a tractable version of this objective function.

**Theorem B.4.** (*Relaxation of unsupervised NRM without noise*) Assume that  $\pi_y \geq \bar{\gamma}$  for all  $1 \leq y \leq K$  for some given  $\bar{\gamma} \in (0, 1/2)$ . Denote

$$V_n := \min_{\theta} \frac{1}{n} \sum_{i=1}^n \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0) x_i + \eta(y, z) \right) \right\} \left( \frac{\|x_i - h(y', z'; 0)\|^2}{2} - \log(\pi_{y', z'}) \right)$$

where  $p_{y', z'} = \exp(\eta(y', z') + \log \pi_{y'}) / \left( \sum_{y=1}^K \exp\left(\max_{z \in \mathcal{L}} \eta(y, z) + \log \pi_y\right) \right)$  for all  $(y', z') \in \mathcal{J}$ . For any  $\theta$ , we define

$$(\bar{y}_i, \bar{z}_i) = \arg \min_{(y, z) \in \mathcal{J}} \left\{ \|x_i - h(y, z; 0)\|^2 - 2\eta(y, z) \right\}$$

and

$$(\tilde{y}_i, \tilde{z}_i) = \arg \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0) x_i + \eta(y, z) \right)$$

as  $1 \leq i \leq n$ . Then, the following holds

(a) Upper bound:

$$\begin{aligned} U_n &\leq \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left\{ \left( \frac{\|x_i - h(\tilde{y}_i, \tilde{z}_i; 0)\|^2}{2} - \log(\pi_{\tilde{y}_i, \tilde{z}_i}) \right) + \underbrace{\left( \log \pi_{\bar{y}_i} - \log \pi_{\tilde{y}_i} \right)}_{\text{prior loss}} \right\} + \log |\mathcal{L}| \\ &\leq V_n + \log \left( \frac{1}{\bar{\gamma}} - 1 \right) + \log |\mathcal{L}| \end{aligned}$$

(b) Lower bound:

$$\begin{aligned} U_n &\geq \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left\{ \left( \frac{\|x_i - h(\tilde{y}_i, \tilde{z}_i; 0)\|^2}{2} - \log(\pi_{\tilde{y}_i, \tilde{z}_i}) \right) + \underbrace{\left( \log \pi_{\bar{y}_i} - \log \pi_{\tilde{y}_i} \right)}_{\text{prior loss}} \right. \\ &\quad \left. + \underbrace{\frac{1}{2} \left( \|h(\bar{y}_i, \bar{z}_i; 0)\|^2 - \|h(\tilde{y}_i, \tilde{z}_i; 0)\|^2 \right)}_{\text{norm loss}} \right\} \\ &\geq V_n + \log \left( \frac{\bar{\gamma}}{1 - \bar{\gamma}} \right) + \min_{\theta} \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \left( \|h(\bar{y}_i, \bar{z}_i; 0)\|^2 - \|h(\tilde{y}_i, \tilde{z}_i; 0)\|^2 \right) \end{aligned}$$

Unlike  $U_n$ , the inference with objective function of  $V_n$  is tractable. According to the upper bound and lower bound of  $U_n$  in terms of  $V_n$ , we can use  $V_n$  as a tractable approximation of  $U_n$  for the inference purpose with unsupervised setting of data when the noise is treated to be 0. Therefore, we achieve the conclusion of part (b) of Theorem 3.3 in the main text. The algorithm for determined (local) minima of  $V_n$  is summarized in Algorithm 2.

### B.3 Relaxation of non-negativity assumption with rendered images

It is clear that the inference with  $V_n$  relies on the non-negativity assumption such that equation (14) holds. Now, we will argue that when the non-negativity assumption with rendered images  $h(y, z; 0)$  does not hold, we can relax  $V_n$  to a more tractable version under that setting.

**Theorem B.5.** *(Relaxation of objective function  $V_n$  when non-negativity assumption does not hold) Assume that  $\pi_y \geq \bar{\gamma}$  for all  $1 \leq y \leq K$  for some given  $\bar{\gamma} \in (0, 1/2)$ . Denote*

$$W_n := \min_{\theta} \frac{1}{n} \sum_{i=1}^n \sum_{y'=1}^K 1_{\left\{y' = \arg \max_{y \in \mathcal{J}} g(y, x_i)\right\}} \left( \frac{\|x_i - g(y', \bar{z}_i)\|^2}{2} - \log(\pi_{y', \bar{z}_i}) \right)$$

where

$$g(y, x) = h^\top(y) \text{MaxPool} \left( \text{ReLU} \left( \text{Conv} \left( \Gamma(\ell), \dots \text{MaxPool} \left( \text{ReLU} \left( \text{Conv} \left( \Gamma(1), I \right) + b(1) \right) \right) \right) \right) \right. \\ \left. \dots + b(\ell) \right)$$

for all  $(x, y)$ . Additionally,  $\bar{z}_i$  is the maximal value of  $z$  in the CNN structure of  $g(y, x_i)$  for  $1 \leq i \leq n$ . For any  $\theta$ , we define

$$(\tilde{y}_i, \tilde{z}_i) = \arg \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0) x_i + \eta(y, z) \right)$$

and  $\bar{y}_i = \arg \max_y g(y, x_i)$  as  $1 \leq i \leq n$ . Then, the following holds

(a) Upper bound:

$$\begin{aligned} V_n &\leq \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left\{ \left( \frac{\|x_i - h(\bar{y}_i, \bar{z}_i; 0)\|^2}{2} - \log(\pi_{\bar{y}_i, \bar{z}_i}) \right) + \underbrace{\left( \log \pi_{\bar{y}_i} - \log \pi_{\tilde{y}_i} \right)}_{\text{prior loss}} \right. \\ &\quad \left. + \frac{1}{2} \underbrace{\left( \|h(\tilde{y}_i, \tilde{z}_i; 0)\|^2 - \|h(\bar{y}_i, \bar{z}_i; 0)\|^2 \right)}_{\text{norm loss}} \right\} \\ &\leq W_n + \log \left( \frac{1}{\bar{\gamma}} - 1 \right) + \min_{\theta} \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \left( \|h(\tilde{y}_i, \tilde{z}_i; 0)\|^2 - \|h(\bar{y}_i, \bar{z}_i; 0)\|^2 \right) \end{aligned}$$



---

**Algorithm 2** Relaxation of unsupervised NRM without noise

---

**Input:** Data  $x_i$ , translation matrices  $T(t; \ell)$ , zero padding matrices  $B(\ell)$ , number of labels  $K$ , number of layers  $L$ .

**Output:** Parameters  $\theta$ .

Initialize  $\theta = \left( \{\mu(y)\}_{y=1}^K, \{\Gamma(\ell)\}_{\ell=1}^L, \{\pi_y\}_{y=1}^K, \{b(\ell)\}_{\ell=1}^L \right)$ .

**while**  $\theta$  has not converged **do**

1. E-Step: Update labels  $(y, z)$  of each data

**for**  $i = 1$  **to**  $n$  **do**

$$(\hat{y}_i, \hat{z}_i) = \arg \max_{y, z} (h^\top(y, z)x_i + \log(\pi_{y, z})).$$

**end for**

2. M-Step: By using Stochastic Gradient Descent (SGD), update  $\theta$  that minimizes

$$\sum_{i=1}^n \left( \frac{\|x_i - h(\hat{y}_i, \hat{z}_i)\|^2}{2} - \log(\pi_{\hat{y}_i, \hat{z}_i}) \right).$$

**end while**

---

(b) Lower bound:

$$\begin{aligned} V_n &\geq \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left\{ \left( \frac{\|x_i - h(\bar{y}_i, \bar{z}_i; 0)\|^2}{2} - \log(\pi_{\bar{y}_i, \bar{z}_i}) \right) + \underbrace{\left( \log \pi_{\bar{y}_i} - \log \pi_{\tilde{y}_i} \right)}_{\text{prior loss}} \right\} \\ &\quad + \underbrace{\frac{1}{2} \left( \|h(\tilde{y}_i, \tilde{z}_i; 0)\|^2 - \|h(\bar{y}_i, \bar{z}_i; 0)\|^2 \right)}_{\text{norm loss}} + \underbrace{\left( g(\bar{y}_i, x_i) - \left\{ h(\tilde{y}_i, \tilde{z}_i; 0)^\top x_i + \eta(\tilde{y}_i, \tilde{z}_i) \right\} \right)}_{\text{CNN loss}} \Bigg\} \\ &\geq W_n + \log \left( \frac{\bar{\gamma}}{1 - \bar{\gamma}} \right) + \min_{\theta} \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \left( \|h(\tilde{y}_i, \tilde{z}_i; 0)\|^2 - \|h(\bar{y}_i, \bar{z}_i; 0)\|^2 \right) \\ &\quad + \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left( g(\bar{y}_i, x_i) - \left\{ h(\tilde{y}_i, \tilde{z}_i; 0)^\top x_i + \eta(\tilde{y}_i, \tilde{z}_i) \right\} \right) \end{aligned}$$

The proof argument of the above theorem is similar to that of Theorem B.4; therefore, it is omitted. The upper bound and lower bound of  $V_n$  in terms of  $W_n$  in Theorem B.5 implies that we can use  $W_n$  as a relaxation of  $V_n$  when the non-negativity assumption with rendered images  $h(y, z; 0)$  does not hold. The algorithm for achieving the (local) minima of  $W_n$  is similar to Algorithm 2.

## B.4 NRM with (semi)-supervised setting

In this section, we consider the application of NRM to the (semi)-supervised setting of the data. Under that setting, only a (full) portion of labels of data  $x_1, \dots, x_n$  is available. Without loss of generality, we also assume that the rendering path  $h(y, z; 0)$  satisfies the non-negativity assumption. For the case that  $h(y, z; 0)$  does not satisfy this assumption, we can argue in the same fashion as that of Theorem B.5. Now, we assume that only the labels  $(y_{n_1+1}, \dots, y_n)$  are unknown for some  $n_1 \geq 0$ . When  $n_1 = 0$ , we have the supervised setting of data while we have the semi-supervised setting of data when  $n - n_1$  is small. Our goal is to build a semi-supervised model based on NRM

such that the clustering information from data  $x_1, \dots, x_{n_1}$  can be used efficiently to increase the accuracy of classifying the labels of data  $x_{n_1+1}, \dots, x_n$ . For the sake of simple inference with that purpose, we only consider the setting of NRM when the noise goes to 0. Our idea of constructing the semi-supervised model based on NRM is inspired by an approximation of the upper bound of maximizing the conditional log likelihood of NRM in terms of the cross entropy and reconstruction loss in part (b) of Theorem 3.3. In particular, we combine the tractable version of reconstruction loss from the unsupervised setting in Theorem 3.3b and the cross entropy of approximate posterior in Theorem 3.3a, which can be formulated as follows

$$\begin{aligned} \min_{\theta} \frac{\alpha_{\text{RC}}}{n} & \left\{ \sum_{i=1}^{n_1} \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0)x_i + \tau(\theta_{y, z}) \right) \right\} \left( \frac{\|x_i - h(y', z'; 0)\|^2}{2} - \log(\pi_{y', z'}) \right) \right. \\ & + \left( \sum_{i=n_1+1}^n \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( h^\top(y_i, z; 0)x_i + \tau(\theta_{y_i, z}) \right) \right\} \left( \frac{\|x_i - h(y_i, z'; 0)\|^2}{2} - \log(\pi_{y_i, z'}) \right) \right\} \\ & - \frac{\alpha_{\text{CE}}}{n - n_1} \sum_{i=n_1+1}^n \log q_\theta(y_i | x_i) \end{aligned}$$

where  $\alpha_{\text{RC}}$  and  $\alpha_{\text{CE}}$  are non-negative weights associated with reconstruction loss and cross entropy respectively. Additionally, the approximate posterior  $q_\theta(y|x_i)$  is chosen as

$$q_\theta(y|x_i) = \frac{\exp \left( \max_{z \in \mathcal{L}} \left\{ h^\top(y, z; 0)x_i + \tau(\theta_{y, z}) \right\} + \log \pi_y \right)}{\sum_{y'=1}^K \exp \left( \max_{z \in \mathcal{L}} \left\{ h^\top(y', z; 0)x_i + \tau(\theta_{y', z}) \right\} + \log \pi_{y'} \right)}.$$

Note that, since the labels  $(y_{n_1+1}, \dots, y_n)$  are known, the reconstruction loss for clustering data  $x_{n_1+1}, \dots, x_n$  in the above objective function indeeds incorporate these information to improve the accuracy of estimating the parameters. We call the above objective function to be *(semi)-supervised NRM without noise*.

**Boosting the accuracy of (semi)-supervised NRM without noise:** In practice, it may happen that the accuracy of classifying data by using the parameters from (semi)-supervised NRM without noise is not very high. To account for that problem, we consider the following general version of (semi)-supervised NRM without noise that includes the variational inference term and

the moment matching term

$$\begin{aligned}
\min_{\theta} \frac{\alpha_{\text{RC}}}{n} & \left\{ \sum_{i=1}^{n_1} \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0) x_i + \tau(\theta_{y, z}) \right) \right\} \left( \frac{\|x_i - h(y', z'; 0)\|^2}{2} - \log(\pi_{y', z'}) \right) \right. \\
& + \left( \sum_{i=n_1+1}^n \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( h^\top(y_i, z; 0) x_i + \tau(\theta_{y_i, z}) \right) \right\} \left( \frac{\|x_i - h(y_i, z'; 0)\|^2}{2} - \log(\pi_{y_i, z'}) \right) \right\} \\
& - \frac{\alpha_{\text{CE}}}{n - n_1} \sum_{i=n_1+1}^n \log q_\theta(y_i | x_i) + \frac{\alpha_{\text{KL}}}{n} \sum_{i=1}^n \sum_{y=1}^K q_\theta(y | x_i) \log \left( \frac{q_\theta(y | x_i)}{\pi_y} \right) \\
& + \alpha_{\text{MM}} \sum_{\ell=1}^L \text{D}_{\text{KL}} \left( \mathcal{N}(\mu_{h(\ell)}, \sigma_{h(\ell)}^2) \parallel \mathcal{N}(\mu_{\psi(\ell)}, \sigma_{\psi(\ell)}^2) \right). \tag{16}
\end{aligned}$$

Here,  $\alpha_{\text{KL}}$  and  $\alpha_{\text{MM}}$  are non-negative weights associated with the variational inference loss and moment matching loss respectively. Additionally,  $\mu_{h(\ell)}, \sigma_{h(\ell)}^2, \mu_{\psi(\ell)}, \sigma_{\psi(\ell)}^2$  in the moment matching loss are defined as follows:

$$\begin{aligned}
\mu_{h(\ell)} &= \frac{1}{n} \sum_{i=1}^n \hat{h}(\ell)_i, & \sigma_{h(\ell)}^2 &= \frac{1}{n} \sum_{i=1}^n (\hat{h}(\ell)_i - \mu_{h(\ell)})^2 \\
\mu_{\psi(\ell)} &= \frac{1}{n} \sum_{i=1}^n \psi(\ell)_i, & \sigma_{\psi(\ell)}^2 &= \frac{1}{n} \sum_{i=1}^n (\psi(\ell)_i - \mu_{\psi(\ell)})^2 \tag{17}
\end{aligned}$$

where  $\hat{h}(\ell)_i$  is the estimated value of  $h(\ell)$  given the optimal latent variables  $\hat{s}^{(\ell)}$  and  $\hat{t}^{(\ell)}$  inferred from the image  $x_i$  for  $1 \leq i \leq n$ . It is clear that when  $\alpha_{\text{KL}} = \alpha_{\text{MM}} = 0$ , we return to (semi)-supervised NRM without noise. In Appendix C, we provide careful theoretical analyses regarding statistical guarantees of model (16).

Now, we will provide heuristic explanations about the improvement in terms of performance of model (16) based on the variational inference term and the moment matching term.

**Regarding the variational term:** The DRMM inference algorithm developed thus far ignores uncertainty in the latent nuisance posterior  $p(y, z | x)$  due to the max-marginalization over  $(y, z)$  in the E-step bottom-up inference. We would like to properly account for this uncertainty for two main reasons: (i) our fundamental hypothesis is that the brain performs probabilistic inference and (ii) uncertainty accounting is very important for good generalization in the semi-supervised setting since we have very little labeled data.

One approach attempts to approximate the true class posterior  $p(y | x)$  for the DRMM. We employ *variational inference*, a technique that enables the approximate inference of the latent posterior. Mathematically, for the DRMM this means we would like to approximate the true class posterior  $p(y | x) \approx q(y | x)$ , where the approximate posterior  $q$  is restricted to some tractable family of distributions (e.g. Gaussian or Categorical). We strategically choose the tractable family to be  $q(y | x) \equiv p(y | \hat{z}, x)$ , where  $\hat{z} \equiv \arg \max_z p(y, z | x)$ . In other words, we choose  $q$  to be restricted to the DRMM family of nuisance max-marginalized class posteriors. Note that this is indeed an approximation, since the true DRMM class posterior has nuisances that are *sum*-marginalized out  $p(y | x) = \sum_z p(y, z | x)$ , whereas the approximating variational family has nuisances that are *max*-marginalized out.

Given our choice of variational family  $q$ , we derive the variational term for the loss function, starting from the principled goal of minimizing the KL-distance  $D_{KL}[q(y|x)||p(y|x)]$  between the true and approximate posteriors with respect to the parameters of  $q$ . As a result, such an optimized  $q$  will tilt towards better approximating  $p(y|x)$ , which in turn means that it will account for *some* of the uncertainty in  $p(z|x)$ . The variational terms in the loss are defined as [53]:

$$\begin{aligned}\mathcal{L}_{VI} &\equiv \mathcal{L}_{RC} + \beta_{KL}\mathcal{L}_{KL} \\ &\equiv -\mathbb{E}_q[\ln p(x|y)] + \beta_{KL}D_{KL}[q(y|x)||p(y)].\end{aligned}\tag{18}$$

This term is quite similar to that used in variational autoencoders (VAE) [9], except for two key differences: (i) here the latent variable  $y$  is discrete categorical rather than continuous Gaussian and (ii) we have employed a slight relaxation of the VAE by allowing for a penalty parameter  $\beta_{KL} \neq 1$ . The latter is motivated by recent experimental results showing that such freedom enables optimal disentangling of the true intrinsic latent variables from the data.

**Regarding the moment matching term:** Batch Normalization can potentially be derived by normalizing the intermediate rendered images  $h(\ell)$ ,  $\ell = 1, 2, \dots, L$  in the NRM by subtracting their means and dividing by their standard deviations under the assumption that the means and standard derivations of  $h(\ell)$  are close to those of the activation  $\psi(\ell)$  in the CNNs. From this intuition, in Section B.4 of Appendix A, we introduce the moment-matching loss to improve the performance of the NRM/CNNs trained for semi-supervised learning tasks.

## B.5 Statistical guarantees for (semi)-supervised setting

For the sake of simplicity with proof argument, we only provide detail theoretical analysis for statistical guarantee with the setting of (16) when the moment matching term is skipped. In particular, we are interested in the following (semi)-supervised model

$$\begin{aligned}Y_n &:= \min_{\theta} \frac{\alpha_{RC}}{n} \left\{ \sum_{i=1}^{n_1} \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0)x_i + \tau(\theta_{y, z}) \right) \right\} \right. \\ &\quad \times \left( \frac{\|x_i - h(y', z'; 0)\|^2}{2} - \log(\pi_{y', z'}) \right) \\ &\quad \left. + \left( \sum_{i=n_1+1}^n \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( h^\top(y_i, z; 0)x_i + \tau(\theta_{y_i, z}) \right) \right\} \left( \frac{\|x_i - h(y_i, z'; 0)\|^2}{2} - \log(\pi_{y_i, z'}) \right) \right) \right\} \\ &\quad - \frac{\alpha_{CE}}{n - n_1} \sum_{i=n_1+1}^n \log q_\theta(y_i|x_i) + \frac{\alpha_{KL}}{n} \sum_{i=1}^n \sum_{y=1}^K q_\theta(y|x_i) \log \left( \frac{q_\theta(y|x_i)}{\pi_y} \right)\end{aligned}\tag{19}$$

where the approximate posterior  $q_\theta(y|x_i)$  is chosen as

$$q_\theta(y|x_i) := \frac{\exp \left( \max_{z \in \mathcal{L}} \left\{ h^\top(y, z; 0)x_i + \tau(\theta_{y, z}) \right\} + \log \pi_y \right)}{\sum_{y'=1}^K \exp \left( \max_{z \in \mathcal{L}} \left\{ h^\top(y', z; 0)x_i + \tau(\theta_{y', z}) \right\} + \log \pi_{y'} \right)}.$$

Here,  $\alpha_{RC}$ ,  $\alpha_{CE}$ , and  $\alpha_{KL}$  are non-negative weights associated with reconstruction loss, cross entropy, and variational inference respectively. As being indicated in the formulation of objection function  $Y_n$ , the only difference between  $Y_n$  and (16) is the weight  $\alpha_{MM}$  regarding moment matching loss in (16) is set to be 0. To ease the presentation with theoretical analyses later, we call the objective function with  $Y_n$  to be *partially labeled latent dependence regularized cross entropy (partially labeled LDCE)*.

**Consistency of partially labeled LDCE:** Firstly, we demonstrate that the objective function of partially labeled LDCE enjoys the consistency guarantee.

**Theorem B.6.** (*Consistency of objective function of partially labeled LDCE*) Assume that  $n_1$  is a function of  $n$  such that  $n_1/n \rightarrow \bar{\lambda}$  as  $n \rightarrow \infty$ . Furthermore,  $\mathbb{P}(\|x\| \leq R) = 1$  as  $x \sim P$  for some given  $R > 0$ . We denote the population version of partially labeled LDCE as follows

$$\begin{aligned} \bar{Y} := & \min_{\theta} \alpha_{RC} \left\{ \bar{\lambda} \left( \int \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0)x + \tau(\theta_{y, z}) \right) \right\} \left( \frac{\|x - h(y', z'; 0)\|^2}{2} \right. \right. \right. \\ & \left. \left. - \log(\pi_{y', z'}) \right) dP(x) \right) + (1 - \bar{\lambda}) \left( \int \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( h^\top(y, z; 0)x + \tau(\theta_{y, z}) \right) \right\} \left( \frac{\|x - h(y, z'; 0)\|^2}{2} \right. \right. \\ & \left. \left. - \log(\pi_{y, z'}) \right) dQ(x, c) \right) \right\} - \alpha_{CE} \int \log q_\theta(y|x) dQ(x, c) + \alpha_{KL} \int \sum_{y=1}^K q_\theta(y|x) \log \left( \frac{q_\theta(y|x)}{\pi_y} \right) dP(x). \end{aligned}$$

Then, we obtain that  $Y_n \rightarrow \bar{Y}$  almost surely as  $n \rightarrow \infty$ .

The detail proof of Theorem B.6 is deferred to Appendix C. Now, we denote

$$\tilde{\theta} := \left( \{\tilde{\mu}(y)\}_{y=1}^K, \{\tilde{\Gamma}(\ell)\}_{\ell=1}^L, \{\tilde{\pi}_y\}_{y=1}^K, \{\tilde{b}(\ell)\}_{\ell=1}^L \right)$$

the optimal solutions of objective function (19). Note that, the existence of these optimal solutions is guaranteed due to the compactness assumption of the parameter spaces  $\Theta_\ell$ ,  $\Omega$ , and  $\Xi_l$  for  $1 \leq \ell \leq L$ . The optimal solutions  $\{\tilde{\mu}(y)\}_{y=1}^K$  and  $\{\tilde{\Gamma}(\ell)\}_{\ell=1}^L$  lead to corresponding set of optimal rendered images  $\tilde{S}_n$ . Similar to the case of SPLD regularized K-means, our goal is to guarantee the consistency of  $\tilde{S}_n$  as well as  $\{\tilde{\pi}_y\}_{y=1}^K, \{\tilde{b}(\ell)\}_{\ell=1}^L$ .

In particular, we denote  $\tilde{\mathcal{F}}_0$  the set of all optimal solutions  $\tilde{\theta}^0$  of population partially labeled LDCE where  $\tilde{\theta}^0 := \left( \{\tilde{\mu}^0(y)\}_{y=1}^K, \{\tilde{\Gamma}_0(\ell)\}_{\ell=1}^L, \{\tilde{\pi}_y^0\}_{y=1}^K, \{\tilde{b}_0(\ell)\}_{\ell=1}^L \right)$ . For each  $\tilde{\theta}^0 \in \tilde{\mathcal{F}}_0$ , we define  $\tilde{S}_0$  the set of optimal rendered images associated with  $\tilde{\theta}^0$ . We denote  $\mathcal{G}(\tilde{\mathcal{F}}_0)$  the corresponding set of all optimal rendered images  $\tilde{S}_0$ , optimal prior probabilities  $\{\tilde{\pi}_y^0\}_{y=1}^K$ , and optimal biases  $\{\tilde{b}_0(\ell)\}_{\ell=1}^L$ .

**Theorem B.7.** (*Consistency of optimal rendering paths and optimal solutions of partially labeled LDCE*) Assume that  $\mathbb{P}(\|x\| \leq R) = 1$  as  $x \sim P$  for some given  $R > 0$ . Then, we obtain that

$$\inf_{(\tilde{S}_0, \{\tilde{\pi}_y^0\}, \{\tilde{b}_0(\ell)\}) \in \mathcal{G}(\tilde{\mathcal{F}}_0)} \left\{ H(\tilde{S}_n, \tilde{S}_0) + \sum_{y=1}^K |\tilde{\pi}_y - \tilde{\pi}_y^0| + \sum_{\ell=1}^L \|\tilde{b}(\ell) - \tilde{b}_0(\ell)\| \right\} \rightarrow 0$$

almost surely as  $n \rightarrow \infty$ .

The detail proof of Theorem B.7 is postponed to Appendix C.

## B.6 Generalization bound for classification framework with (semi)-supervised setting

In this section, we provide a simple generalization bound for certain classification function with the optimal solutions  $\tilde{\theta} = \left( \{\tilde{\mu}(y)\}_{y=1}^K, \{\tilde{\Gamma}(\ell)\}_{\ell=1}^L, \{\tilde{\pi}_y\}_{y=1}^K, \{\tilde{b}(\ell)\}_{\ell=1}^L \right)$  of (16). In particular, we denote the following function  $f : \mathbb{R}^{D^{(0)}} \times \{1, \dots, K\} \rightarrow \mathbb{R}$  as

$$f(x, y) = \max_{z \in \mathcal{L}} \left\{ \tilde{h}^\top(y, z; 0)x + \tau(\tilde{\theta}_{y,z}) \right\} + \log \tilde{\pi}_y$$

for all  $(x, y) \in \mathbb{R}^{D^{(0)}} \times \{1, \dots, K\}$  where

$$\begin{aligned} \tilde{h}(y, z; 0) &:= \tilde{\Lambda}(z; 1) \dots \tilde{\Lambda}(z; L) \tilde{\mu}(y), \\ \tilde{\Lambda}(z; \ell) &:= \sum_{p \in \mathcal{P}(\ell)} s(\ell, p) T(t; \ell, p) B(\ell, p) \tilde{\Gamma}(\ell, p) \end{aligned}$$

for all  $(y, z)$  and  $1 \leq \ell \leq L$ . To achieve the generalization bound regarding that classification function, we rely on the study of generalization bound with margin loss. For the simplicity of argument, we assume that the true labels of  $x_1, \dots, x_n$  are  $y_1, \dots, y_n$  while  $y_1, \dots, y_{n_1}$  are not available to train. The margin of a labeled example  $(x, y)$  based on  $f$  can be defined as

$$\rho(f, x, y) = f(x, y) - \max_{l \neq y} f(x, l).$$

Therefore, the classification function  $f$  misspecifies the labeled example  $(x, y)$  as long as  $\rho(f, x, y) \leq 0$ . The empirical margin error of  $f$  at margin coefficient  $\Gamma \geq 0$  is

$$R_{n,\gamma}(f) = \frac{1}{n} \sum_{i=1}^n 1_{\{\rho(f, x_i, y_i) \leq \gamma\}}.$$

It is clear that  $R_{n,0}(f)$  is the empirical risk of 0-1 loss, i.e., we have

$$R_{n,0}(f) = \frac{1}{n} \sum_{i=1}^n 1_{\left\{ \arg \max_{1 \leq y \leq K} f(x_i, y) \neq y_i \right\}}.$$

Similar to the argument in the case of SPLD regularized K-means, the optimal solutions  $\tilde{\theta}$  of partially labeled LDCE lead to a set of rendered images  $\tilde{h}(y, z; 0)$  for all  $(y, z) \in \mathcal{J}$ . However, only a small fraction of rendering paths are indeed active in the following sense. There exists a subset  $\mathcal{L}_n$  of  $\mathcal{L}$  such that  $|\mathcal{L}_n| \leq \bar{\tau}_n |\mathcal{L}|$  where  $\bar{\tau}_n \in (0, 1]$ , which is independent of data  $(x_1, y_1), \dots, (x_n, y_n)$ , and the following holds

$$\max_{z \in \mathcal{L}} \left\{ \tilde{h}^\top(y, z; 0)x + \tau(\tilde{\theta}_{y,z}) \right\} + \log \tilde{\pi}_y = \max_{z \in \mathcal{L}_n} \left\{ \tilde{h}^\top(y, z; 0)x + \tau(\tilde{\theta}_{y,z}) \right\} + \log \tilde{\pi}_y$$

for all  $1 \leq y \leq K$ . The above equation implies that

$$R_{n,\Gamma}(f) = R_{n,\gamma}(f_{\bar{\gamma}_n})$$

for all  $\Gamma \geq 0$  and  $n \geq 1$  where  $f_{\bar{\gamma}_n}(x, y) = \max_{z \in \mathcal{L}_n} \left\{ \tilde{h}^\top(y, z; 0)x + \tau(\tilde{\theta}_{y,z}) \right\} + \log \tilde{\pi}_y$  for all  $(x, y)$ . With that connection, we denote the expected margin error of classification function  $f_{\bar{\gamma}_n}$  at margin coefficient  $\Gamma \geq 0$  is

$$R_\gamma(f_{\bar{\gamma}_n}) = \mathbb{E}1_{\{\rho(f_{\bar{\gamma}_n}, x, y) \leq \gamma\}}.$$

The generalization bound that we establish in this section will base on the gap between the expected margin error  $R_0(f_{\bar{\gamma}_n})$  and its corresponding empirical version  $R_{n,\Gamma}(f_{\bar{\gamma}_n})$ , which is also  $R_{n,\Gamma}(f)$ .

**Theorem B.8.** (*Generalization bound for margin-based classification*) Assume that  $P(\|x\| \leq R) = 1$  for some given  $R > 0$  and  $x \sim P$ . Additionally, the parameter spaces  $\Theta_\ell$  and  $\Omega$  are chosen such that  $\|h(y, z; 0)\| \leq R$  for all  $(y, z) \in \mathcal{J}$ . For any  $\delta > 0$ , with probability at least  $1 - \delta$ , we have

$$\begin{aligned} R_0(f_{\bar{\gamma}_n}) \leq & \inf_{\gamma \in (0,1]} \left\{ R_{n,\gamma}(f_{\bar{\gamma}_n}) + \frac{8K(2K-1)}{\Gamma\sqrt{n}} \left( 2\bar{\gamma}_n |\mathcal{L}|(R^2 + 1) + |\log \bar{\gamma}| \right) \right. \\ & \left. + \left( \frac{\log \log_2(2\gamma^{-1})}{n} \right)^{1/2} + \sqrt{\frac{\log(2\delta^{-1})}{2n}} \right\} \end{aligned}$$

where  $\bar{\gamma}$  is the lower bound of prior probability  $\pi_y$  for all  $y$ .

*Remark B.9.* The result of Theorem B.8 gives a simple characterization for the generalization bound of classification setup from optimal solutions of partially labeled LDCE based on the number of active rendering paths, which is inherent to the structure of NRM. Such dependence of generalization bound on the number of active rendering paths  $\bar{\gamma}_n |\mathcal{L}|$  is rather interesting and may provide a new perspective on understanding the generalization bound. Nevertheless, there are certain limitations regarding the current generalization gap: (1) the active ratio  $\bar{\gamma}_n$  may change with the sample size unless we put certain constraints on the sparsity of switching variables  $a$  to reduce the number of active optimal rendering paths; (2) the generalization bound is depth-dependent due to the involvement of the number of rendering paths  $|\mathcal{L}|$ . This is mainly because we have not fully taken into account all the structures of CNNs for the studying of generalization bound. Given some current progress on depth-independent generalization bound [41, 42], it is an interesting direction to explore whether the techniques in these work can be employed to improve  $|\mathcal{L}|$  in the generalization bound in Theorem B.8.

## B.7 Neural Rendering Model is the Unifying Framework for Networks in the Convnet Family

The structure of the rendering matrices  $\Lambda(\ell)$  gives rise to MaxPooling, ReLU, and convolution operators in the CNNs. By modifying the structure of  $\Lambda(\ell)$ , we can derive different types of networks in the convolutional neural network family. In this section, we define and explore several other interesting variants of NRM: the Residual NRM (ResNRM) and the Dense NRM (DenseNRM). Inference algorithms in these NRMs yield ResNet [54] and DenseNet [55], respectively. Proofs for

these correspondence are given in Appendix C. Both ResNet and DenseNet are among state-of-the-art neural networks for object recognition and popularly used for other visual perceptual inference tasks. These two architectures employ skip connections (a.k.a., shortcuts) to create short paths from early layers to later layers. During training, the short paths help avoid the vanishing-gradient problem and allow the network to propagate and reuse features.

### B.7.1 Residual Neural Rendering Model Yields ResNet

In a ResNet, layers learn residual functions with reference to the layer inputs. In particular, as illustrated in Fig. 6, layers in a ResNet are reformulated to represent the mapping  $F(\psi) + W_{\text{skip}}\psi$  and the layers try to fit the residual mapping  $F(\psi)$  where  $\psi$  is the input feature. The term  $W_{\text{skip}}\psi$  accounts for the skip connections/shortcuts [54]. In order to derive the ResNet, we rewrite the rendering matrix  $\Lambda(\ell)$  as the sum of a shortcut matrix  $\Lambda_{\text{skip}}(\ell)$  and a rendering matrix, both of which can be updated during the training. The shortcut matrices yields skip connections in [54]. Note that  $\Lambda_{\text{skip}}(\ell)$  depends on the template selecting latent variables  $s(\ell)$ . In the rest of this section, for clarity, we will refer to  $\Lambda(\ell)$  and  $\Lambda_{\text{skip}}(\ell)$  as  $\Lambda(t, s; \ell)$  and  $\Lambda_{\text{skip}}(s; \ell)$ , respectively, to show their dependency on latent variables in NRM. We define the Residual Neural Rendering Model as follows:

**Definition B.10.** The Residual Neural Rendering Model (ResNRM) is the Neural Rendering Model whose rendering process from layer  $\ell$  to layer  $\ell - 1$ , for some  $\ell \in \{1, 2, \dots, L\}$ , has the residual form as follows:

$$h(\ell - 1) := (\Lambda(t, s; \ell) + \Lambda_{\text{skip}}(s; \ell)) h(\ell), \quad (20)$$

where  $\Lambda_{\text{skip}}(t, s; \ell)$  is the shortcut matrices that results in skip connections in the corresponding ResNet. In particular,  $\Lambda_{\text{skip}}(t, s; \ell)$  has the following form:

$$\Lambda_{\text{skip}}(s; \ell) = \tilde{\Lambda}_{\text{skip}}(\ell) M(s; \ell), \quad (21)$$

where  $M(s; \ell) \equiv \text{diag}(s(\ell)) \in \mathbb{R}^{D(\ell) \times D(\ell)}$  is a diagonal matrix whose diagonal is the vector  $s(\ell)$ . This matrix selects the templates for rendering. Furthermore,  $\tilde{\Lambda}_{\text{skip}}(\ell)$  is a rendering matrix that is independent of latent variables  $t$  and  $s$ .

The following theorem show that similar to how CNNs can be derived from NRM, ResNet can be derived as a bottom-up inference in ResNRM.

**Theorem B.11.** Inference in ResNRM yields skip connetions. In particular, if the rendering process at layer  $\ell$  has the residual form as in Definition B.10, the inference at this layer takes the following form:

$$\begin{aligned} \psi(\ell) &\equiv \max_{t(\ell), s(\ell)} \left\{ \left( \Lambda^\top(t, s; \ell) + \Lambda_{\text{skip}}^\top(s; \ell) \right) \psi(\ell - 1) + b(\ell) \right\} \\ &= \text{MaxPool} \left( \text{ReLU} \left( \text{Conv}(\Gamma^\top(\ell), \psi(\ell - 1)) + b(\ell) + \underbrace{\tilde{\Lambda}_{\text{skip}}^\top(\ell) \psi(\ell - 1)}_{\text{skip connection}} \right) \right) \\ &\stackrel{d}{=} \text{MaxPool} \left( \text{ReLU} \left( \text{Conv}(W(\ell), \psi(\ell - 1)) + b(\ell) + \underbrace{W_{\text{skip}}(\ell) \psi(\ell - 1)}_{\text{skip connection}} \right) \right). \end{aligned} \quad (22)$$



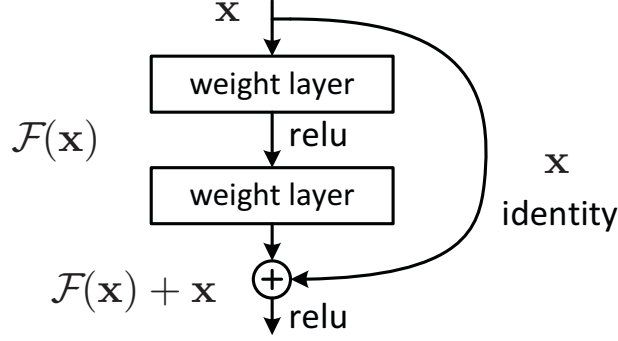


Figure 6: ResNet building block as in [54]

Here, when  $\psi(\ell - 1)$  and  $\psi(\ell)$  have the same dimensions,  $\tilde{\Lambda}_{\text{skip}}(\ell)$  is chosen to be an constant identity matrix in order to derive the parameter-free, identity shortcut among layers of the same size in the ResNet. When  $\psi(\ell - 1)$  and  $\psi(\ell)$  have the different dimensions,  $\tilde{\Lambda}_{\text{skip}}(\ell)$  is chosen to be a learnable shortcut matrix which yields the projection shortcut  $W_{\text{skip}}(\ell)$  among layers of different sizes in the ResNet. As mentioned above, identity shortcuts and projection shortcuts are two types of skip connections in the ResNet. The operator  $\stackrel{d}{=}$  implies that discriminative relaxation is applied.

In practice, the skip connections are usually across two layers or three layers. This is indeed a straightforward extension from the ResNRM. In particular, the ResNRM building block that corresponds to the ResNet building block in [54] (see Fig. 6) takes the following form:

$$h(\ell - 2) := (\Lambda(t, s; \ell - 1)\Lambda(t, s; \ell) + \Lambda_{\text{skip}}(s; \ell)) h(\ell).$$

In inference, this ResNRM building block yields the ResNet building block in Fig. 6:

$$\begin{aligned} \psi(\ell) &= \text{ReLU} \left( \text{Conv} \left( \Gamma^\top(\ell), \text{ReLU} \left( \text{Conv} \left( \Gamma^\top(\ell - 1), \psi(\ell - 2) \right) + b(\ell - 1) \right) \right) \right. \\ &\quad \left. + b(\ell) + \tilde{\Lambda}_{\text{skip}}^\top(\ell) \psi(\ell - 2) \right) \\ &\stackrel{d}{=} \text{ReLU} \left( \text{Conv} \left( W(\ell), \text{ReLU} \left( \text{Conv} \left( W(\ell - 1), \psi(\ell - 2) \right) + b(\ell - 1) \right) \right) \right. \\ &\quad \left. + b(\ell) + W_{\text{skip}}(\ell) \psi(\ell - 2) \right). \end{aligned} \tag{23}$$

### B.7.2 Dense Neural Rendering Model Yields DenseNet

In a DenseNet [55], instead of combining features through summation, the skip connections concatenate features. In addition, within a building block, all layers are connected to each other (see Fig. 7). Similar to how ResNet can be derived from ResNRM, DenseNet can also be derived from a variant of NRM, which we call the Dense Neural Rendering Model (DenseNRM). In DenseNRM, the rendering matrix  $\Lambda(\ell)$  is concatenated by an identity matrix. This extra identity matrix, in inference, yields the skip connections that concatenate features at different layers in a DenseNet. We define DenseNRM as follows.

**Definition B.12.** The Dense Neural Rendering Model (DenseNRM) is the Neural Rendering Model whose rendering process from layer  $\ell$  to layer  $\ell - 1$ , for some  $\ell \in \{1, 2, \dots, L\}$ , has the residual form as follows:

$$h(\ell - 1) := [\Lambda(t, s; \ell)h(\ell), \mathbf{1}_{D(\ell)}h(\ell)] . \quad (24)$$

We again denote  $\Lambda(\ell)$  as  $\Lambda(t, s; \ell)$  to show the dependency of  $\Lambda(\ell)$  on the latent variables  $t(\ell)$  and  $s(\ell)$ . The following theorem establishes the connections between DenseNet and DenseNRM.

**Theorem B.13.** Inference in DenseNRM yields DenseNet building blocks. In particular, if the rendering process at layer  $\ell$  has the dense form as in Definition B.12, the inference at this layer takes the following form:

$$\begin{aligned} \psi(\ell) &\equiv \begin{bmatrix} \max_{t(\ell), s(\ell)} \{ \Lambda^\top(t, s; \ell) \psi(\ell - 1) + b(\ell) \} \\ \psi(\ell - 1) \end{bmatrix} \\ &= \begin{bmatrix} \text{MaxPool}(\text{ReLU}(\text{Conv}(\Gamma^\top(\ell), \psi(\ell - 1)) + b(\ell))) \\ \psi(\ell - 1) \end{bmatrix} \\ &\stackrel{d}{=} \begin{bmatrix} \text{MaxPool}(\text{ReLU}(\text{Conv}(W(\ell), \psi(\ell - 1)) + b(\ell))) \\ \psi(\ell - 1) \end{bmatrix} . \end{aligned} \quad (25)$$

In Eqn. 25, we concatenate the output  $\text{MaxPool}(\text{ReLU}(\text{Conv}(W(\ell), \psi(\ell - 1)) + b(\ell)))$  at layer  $\ell$  with the input feature  $\psi(\ell - 1)$  at layer  $\ell - 1$  to generate the input to the next layer  $\psi(\ell)$ , just like in the DenseNet. Proofs for Theorem B.11 and B.13 can be found in Appendix B. The approach to proving Theorem B.11 can be used to prove the result in Eqn. 23.

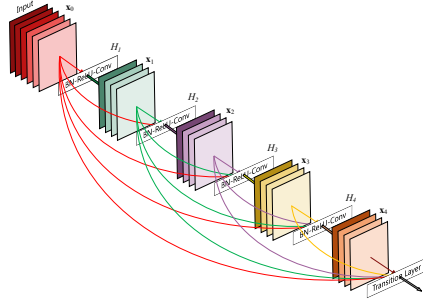


Figure 7: DenseNet building block as in [55]

## C Appendix C

In this appendix, we provide the proofs for key results in the main text as well as Appendix A and Appendix B.

### C.1 Proof for Theorem 3.2: Deriving Convolutional Neural Networks from the Neural Rendering Model

To ease the clarity of the proof presentation, we ignore the normalizing factor  $\frac{1}{\sigma^2}$  and only consider the proof for two layers, i.e.,  $L = 2$ . The argument for  $L \geq 3$  is similar and can be derived recursively from the proof for  $L = 2$ . Similar proof holds when the normalizing factor  $\frac{1}{\sigma^2}$  is considered. Now, we obtain that

$$\begin{aligned}
& \max_z \left\{ h^\top(y, z; 0)x + \eta(y, z) \right\} \\
&= \max_{\substack{t(1), s(1) \\ t(2), s(2)}} \left\{ \sum_{p_1 \in \mathcal{P}(1)} h(1, p_1) s(1, p_1) \Gamma^\top(1, p_1) B^\top(1, p_1) T^\top(t; 1, p_1) x \right. \\
&\quad \left. + \sum_{p_1 \in \mathcal{P}(1)} h(1, p_1) s(1, p_1) b(t; 1, p_1) + \sum_{p_2 \in \mathcal{P}(2)} \mu(y; p_2) s(2, p_2) b(t; 2, p_2) \right\} \\
&= \max_{\substack{t(1), s(1) \\ t(2), s(2)}} \left\{ \sum_{p_1 \in \mathcal{P}(1)} h(1, p_1) s(1, p_1) \left( \Gamma^\top(1, p_1) B^\top(1, p_1) T^\top(t; 1, p_1) x + b(t; 1, p_1) \right) \right. \\
&\quad \left. + \sum_{p_2 \in \mathcal{P}(2)} \mu(y; p_2) s(2, p_2) b(t; 2, p_2) \right\} \\
&\stackrel{(a)}{=} \max_{t(2), s(2)} \left\{ \sum_{p_1 \in \mathcal{P}(1)} h(1, p_1) \max_{t(1, p_1), s(1, p_1)} s(1, p_1) \left( \Gamma^\top(1, p_1) B^\top(1, p_1) T^\top(t; 1, p_1) x + b(t; 1, p_1) \right) \right. \\
&\quad \left. + \sum_{p_2 \in \mathcal{P}(2)} \mu(y; p_2) s(2, p_2) b(t; 2, p_2) \right\} = A,
\end{aligned}$$

where equation in (a) is due to the non-negativity assumption that  $h(1, p_1) \geq 0$ , just as in max-sum and max-product message passing. We define  $\psi(1, p_1)$  as follows:

$$\psi(1, p_1) = \max_{t(1, p_1), s(1, p_1)} s(1, p_1) \left( \Gamma^\top(1, p_1) B^\top(1, p_1) T^\top(t; 1, p_1) x + b(t; 1, p_1) \right),$$

and let  $\psi(1) = (\psi(1, p_1))_{p_1 \in \mathcal{P}(1)}$  be the vector of  $\psi(1, p_1)$ . The following holds:

$$\begin{aligned}
A &= \max_{t(2), s(2)} \left\{ h^\top(1) \psi(1) + \sum_{p_2 \in \mathcal{P}(2)} \mu(y; p_2) s(2, p_2) b(t; 2, p_2) \right\} \\
&\stackrel{(b)}{=} \max_{t(2), s(2)} \left\{ \sum_{p_2 \in \mathcal{P}(2)} \mu(y; p_2) s(2, p_2) \Gamma^\top(2, p_2) B^\top(2, p_2) T^\top(t; 2, p_2) \psi(1) \right. \\
&\quad \left. + \sum_{p_2 \in \mathcal{P}(2)} \mu(y; p_2) s(2, p_2) b(t; 2, p_2) \right\} \\
&\stackrel{(c)}{=} \sum_{p_2 \in \mathcal{P}(2)} \mu(y; p_2) \underbrace{\max_{t(2), s(2)} s(2, p_2) \left( \Gamma^\top(2, p_2) B^\top(2, p_2) T^\top(t; 2, p_2) \psi(1) + b(t; 2, p_2) \right)}_{\psi(2, p_2)} \\
&= \mu^\top(y) \psi(2).
\end{aligned}$$

Here  $\psi(2) = (\psi(2, p_2))_{p_2 \in \mathcal{P}(2)}$  is the vector of  $\psi(2, p_2)$ , and in line (b) we substitute  $h(1)$  by:

$$h(1) = \sum_{p \in \mathcal{P}(2)} s(2, p_2) T(t; 2, p_2) B(2, p_2) \Gamma(2, p_2) \mu(y; p_2).$$

Notice that line (a) and (c) form a recursion. Therefore, we finish proving that the feedforward step in CNNs is the latent variable inference in NRM if  $\psi(1)$  has the structure of the building block of CNNs, i.e.  $\text{MaxPool}(\text{ReLU}(\text{Conv}()))$ . Indeed,

$$\begin{aligned}
\psi(1) &= (\psi(1, p(1)))_{p(1) \in \mathcal{P}(1)} \\
&= \left( \max_{t(1, p_1), s(1, p_1)} s(1, p_1) \left( \Gamma^\top(1, p_1) B^\top(1, p_1) T^\top(t; 1, p_1) x + b(t; 1, p_1) \right) \right)_{p(1) \in \mathcal{P}(1)} \\
&= \left( \max \left( \left( \text{ReLU} \left( \Gamma^\top(1, p_1) B^\top(1, p_1) T^\top(t; 1, p_1) x + b(t; 1, p_1) \right) \right)_{t(1, p_1)=0,1,2,3} \right) \right)_{p(1) \in \mathcal{P}(1)} \tag{26} \\
&= \text{MaxPool} \left( \text{ReLU} \left( \text{Conv} \left( \Gamma^\top(1), x \right) + b(t; 1) \right) \right) \\
&\stackrel{d}{=} \text{MaxPool} \left( \text{ReLU} \left( \text{Conv} \left( W(1), x \right) + b(t; 1) \right) \right),
\end{aligned}$$

where  $W(1) = \Gamma^\top(1)$  corresponds to the weights at layer 1 of the CNN, and  $\stackrel{d}{=}$  implies that discriminative relaxation is applied. In Eqn. 26, since  $s(1, p_1) \in \{0, 1\}$ ,  $\max_{s(1, p_1)} (s(1, p_1) \times \cdot)$  is equivalent to  $\max(\cdot, 0)$  and, therefore, yields the ReLU function. Also, in order to compute  $\max_{t(1, p_1)}()$ , we take the

brute force approach, computing the function inside the parentheses for all possible values of  $t(1, p_1)$  and pick the maximum one. This procedure is equivalent to the MaxPool operator in CNNs. Here we can make the bias term  $b(t; 1)$  independent of  $t$  and  $b(t; 1, p_1)$  are the same for all pixels  $p_1$  in the same feature map of  $h(1)$  as in CNNs. Similarly,  $\psi(2) = \text{MaxPool}(\text{ReLU}(\text{Conv}(W(2), \psi(1)) + b(2)))$ . Thus, we obtain the conclusion of the theorem for  $L = 2$ .

## C.2 Proof for Theorem B.11: Deriving the Residual Networks from the Residual Neural Rendering Model

Similar to the proof in Section C.1 above, when the rendering process at each layer in ResNRM is as in Eqn. 20, the activations  $\psi(\ell)$  is given by:

$$\begin{aligned}
\psi(\ell) &= \left( \max_{t(\ell,p), s(\ell,p)} s(\ell,p) \left( \left( \Gamma^\top(\ell,p) B^\top(\ell,p) T^\top(t;\ell,p) + \tilde{\Lambda}_{\text{skip}}^\top(\ell,p) \right) \psi(\ell-1) + b(t;\ell,p) \right) \right)_{p \in \mathcal{P}(\ell)} \\
&= \left( \max \left( \left( \text{ReLU} \left( \Gamma^\top(\ell,p) B^\top(\ell,p) T^\top(t;\ell,p) \psi(\ell-1) + b(t;\ell,p) \right. \right. \right. \right. \\
&\quad \left. \left. \left. + \tilde{\Lambda}_{\text{skip}}^\top(\ell,p) \psi(\ell-1) \right) \right)_{t(\ell,p)=0,1,2,3} \right)_{p \in \mathcal{P}(\ell)} \\
&= \text{MaxPool} \left( \text{ReLU} \left( \text{Conv} \left( \Gamma^\top(\ell,p), \psi(\ell-1) \right) + b(t;\ell) + \tilde{\Lambda}_{\text{skip}}^\top(\ell) \psi(\ell-1) \right) \right) \\
&\stackrel{d}{=} \text{MaxPool} \left( \text{ReLU} \left( \text{Conv}(W^{(\ell)}, \psi(\ell-1)) + b(t;\ell) + \underbrace{W_{\text{skip}}(\ell) \psi(\ell-1)}_{\text{skip connection}} \right) \right). \tag{27}
\end{aligned}$$

Here we can again make the bias term  $b(t;\ell)$  independent of  $t$  and  $b(t;\ell,p)$  are the same for all pixels  $p_\ell$  in the same feature map of  $h(\ell)$  as in CNNs. We obtain the conclusion of the Theorem B.11.

## C.3 Proof for Theorem B.13: Deriving the Densely Connected Networks from the Dense Neural Rendering Model

Similar to the proof for Theorem B.11 above, when the rendering process at each layer in ResNRM is as in Eqn. 24, the activations  $\psi(\ell)$  is given by:

$$\begin{aligned}
\psi(\ell) &\equiv \left[ \begin{array}{c} \left( \max_{s(\ell,p), t(\ell,p)} (s(\ell,p) (\Gamma^\top(\ell,p) B^\top(\ell,p) T^\top(t;\ell,p) \psi(\ell-1) + b(t;\ell,p))) \right)_{p \in \mathcal{P}(\ell)} \\ \psi(\ell-1) \end{array} \right] \\
&= \left[ \begin{array}{c} \left( \max \left( \left( \text{ReLU} \left( \Gamma^\top(\ell,p) B^\top(\ell,p) T^\top(t;\ell,p) \psi(\ell-1) + b(t;\ell,p) \right) \right)_{t(\ell,p)=0,1,2,3} \right)_{p \in \mathcal{P}(\ell)} \\ \psi(\ell-1) \end{array} \right] \\
&= \left[ \begin{array}{c} \text{MaxPool}(\text{ReLU}(\text{Conv}(\Gamma^\top(\ell,p), \psi(\ell-1)) + b(t;\ell))) \\ \psi(\ell-1) \end{array} \right] \\
&\stackrel{d}{=} \left[ \begin{array}{c} \text{MaxPool}(\text{ReLU}(\text{Conv}(W^{(\ell)}, \psi(\ell-1)) + b(t;\ell))) \\ \psi(\ell-1) \end{array} \right]. \tag{28}
\end{aligned}$$

Again we can make the bias term  $b(t;\ell)$  independent of  $t$  and  $b(t;\ell,p)$  are the same for all pixels  $p_\ell$  in the same feature map of  $h(\ell)$  as in CNNs. We obtain the conclusion of the Theorem B.13.

## C.4 Proving that the Parametrized Joint Prior $p(y,z)$ is a Conjugate Prior

Again, for simplicity, we only consider the proof for two layers. The argument for  $L \geq 3$  is similar and can be derived recursively from the proof for  $L = 2$ . In the derivation below,  $h(2)$  is  $\mu(y)$ . As in Eqn.1 in the definition of the NRM, the joint prior of  $y$  and  $z$  is given by:

$$\begin{aligned}
p(y, z) &\propto \exp \left( \frac{1}{\sigma^2} b^\top(t; 2)(s(2) \odot h(2)) + \frac{1}{\sigma^2} b^\top(t; 1)(s(1) \odot h(1)) + \ln \pi_y \right) \\
&= \exp \left( \frac{1}{\sigma^2} h^\top(2)(b(t; 2) \odot s(2)) + \frac{1}{\sigma^2} h^\top(1)(b(t; 1) \odot s(1)) + \ln \pi_y \right) \\
&= \exp \left( \frac{1}{\sigma^2} h^\top(2)(b(t; 2) \odot s(2)) + \frac{1}{\sigma^2} h^\top(2) \Lambda^\top(2)(b(t; 1) \odot s(1)) + \ln \pi_y \right) \\
&= \exp \left( \frac{1}{\sigma^2} h^\top(2) \left[ \Lambda^\top(2)(b(t; 1) \odot s(1)) + b(t; 2) \odot s(2) \right] + \ln \pi_y \right) \tag{29}
\end{aligned}$$

Furthermore, as explained in Section C.7 of Appendix D, due to the constant norm assumption with  $h(y, z; 0)$ , the likelihood  $p(x|y, z)$  is estimated as follows:

$$\begin{aligned}
p(x|y, z) &\propto \exp \left( \frac{1}{\sigma^2} h^\top(y, z; 0)x \right) \\
&= \exp \left( \frac{1}{\sigma^2} h^\top(2) \Lambda^\top(2) \Lambda^\top(1)x \right)
\end{aligned}$$

The posterior  $p(y, z|x)$  is given by:

$$\begin{aligned}
p(y, z|x) &\propto \exp \left( \frac{1}{\sigma^2} h^\top(2) \left[ \Lambda^\top(2)(b(t; 1) \odot s(1)) + \Lambda^\top(2) \Lambda^\top(1)x + b(t; 2) \odot s(2) \right] + \ln \pi_y \right) \\
&= \exp \left( \frac{1}{\sigma^2} h^\top(2) \left[ \Lambda^\top(2)(b(t; 1) \odot s(1) + \Lambda^\top(1)x) + b(t; 2) \odot s(2) \right] + \ln \pi_y \right) \tag{30}
\end{aligned}$$

Comparing Eqn. 29 and Eqn. 30, we see that the prior and the posterior have the same functional form. This completes the proof.

## C.5 Deriving Other Components in the Convolutional Neural Networks

### C.5.1 Deriving the Leaky Rectified Linear Unit (Leaky ReLU)

The Leaky ReLU can be derived from the NRM in the same way as we derive the ReLU in Section C.1, but instead of  $s(\ell) \in \{0, 1\}$ , we now let  $s(\ell) \in \{\alpha, 1\}$ , where  $\alpha$  is a small positive constant. Then, in Eqn. 26,  $\max_{s(\ell, p)}(s(\ell, p) \times \cdot)$  is equivalent to  $[\cdot < 0](\alpha \times \cdot) + [\cdot \geq 0](\cdot)$ , which is the LeakyReLU function. Note that compared to Eqn. 26, here we replace the layer number (1) by  $(\ell)$  since the same derivation can be applied at any layer.

## C.6 Proofs for Batch Normalization

In this section we will derive the batch normalization from a 2-layer DRM. The same derivation can be generalized to the case of K-layer DRM.

In order to derive the Batch Normalization, we normalize the intermediate rendering image  $h(\ell)$ .

$$\begin{aligned}
& h(y, z; 0) \\
&= \sum_{p_1 \in \mathcal{P}(1)} s(1, p_1) T(t; 1, p_1) B(1, p_1) \Gamma(1, p_1) \frac{1}{\sigma_h(1, p_1)} (h(1, p_1) - \mathbb{E}_h(1, p_1)) - \mathbb{E}_{h(y, z; 0)} \\
&= \sum_{p_1 \in \mathcal{P}(1)} s(1, p_1) T(t; 1, p_1) B(1, p_1) \Gamma(1, p_1) \frac{1}{\sigma_h(1, p_1)} \\
&\quad \times \left( \left( \sum_{p_2 \in \mathcal{P}(2)} s(1, p_2) T(t; 2, p_2) B(2, p_2) \Gamma(2, p_2) \frac{1}{\sigma_h(y, z; 0)} h(y, z; 0) \right) (1, p_1) - \mathbb{E}_h(1, p_1) \right) \\
&\quad - \mathbb{E}_{h(y, z; 0)} \tag{31}
\end{aligned}$$

During inference, we de-mean the input image and find  $z^* = \arg \max_z h^\top(y, z; 0)(x - \mathbb{E}_{h(y, z; 0)})$ . In particular, the inference can be derived as follows:

$$\begin{aligned}
& \max_z h^\top(y, z; 0)(x - \mathbb{E}_{h(y, z; 0)}) \\
&= \max_z \left( \sum_{p_1 \in \mathcal{P}(1)} s(1, p_1) T(t; 1, p_1) B(1, p_1) \Gamma(1, p_1) \frac{1}{\sigma_h(1, p_1)} \left( h(1, p_1) - \mathbb{E}_h(1, p_1) \right) \right)^\top (x - \mathbb{E}_{h(y, z; 0)}) \\
&\quad - \underbrace{\mathbb{E}_{h(y, z; 0)}^\top (x - \mathbb{E}_{h(y, z; 0)})}_{\text{const w.r.t } y, z} \\
&\stackrel{a}{\approx} \max_z \sum_{p_1 \in \mathcal{P}(1)} \underbrace{\left( s(1, p_1) \left( h(1, p_1) - \mathbb{E}_h(1, p_1) \right) \frac{1}{\sigma_h(1, p_1)} \right)}_{\text{scalar}} \underbrace{\left( \Gamma^\top(1, p_1) B^\top(1, p_1) T^\top(t; 1, p_1) \right)}_{\text{row vector}} \underbrace{(x - \mathbb{E}_i)}_{\text{column vector}} \\
&\quad \underbrace{\hspace{10em}}_{\text{dot product}} \\
&+ \text{const} := A + \text{const}.
\end{aligned}$$

Direct computation leads to

$$\begin{aligned}
A &= \max_z \sum_{p_1 \in \mathcal{P}(1)} \left( s(1, p_1) \left( h(1, p_1) - \mathbb{E}_h(1, p_1) \right) \underbrace{\frac{\sigma_\psi(1, p_1)}{\sigma_h(1, p_1)}}_{\alpha(1, p_1)} \underbrace{\frac{1}{\sigma_\psi(1, p_1)}}_{\text{normalize}} \right. \\
&\quad \left. \times \underbrace{\left( \Gamma^\top(1, p_1) B^\top(1, p_1) T^\top(t; 1, p_1) x - \mathbb{E}[\Gamma^\top(1, p_1) B^\top(1, p_1) T^\top(t; 1, p_1) x] \right)}_{\text{de-mean}} \right) \\
&= \max_{z(2)} \sum_{p_1 \in \mathcal{P}(1)} \max_{s(1, p_1), t(1, p_1)} \left( s(1, p_1) \left( h(1, p_1) - \mathbb{E}_h(1, p_1) \right) \right. \\
&\quad \left. \times \text{BatchNorm}(\Gamma^\top(1, p_1) B^\top(1, p_1) T^\top(t; 1, p_1) x; \alpha(1, p_1)) \right) \\
&= \max_{z(2)} \left( h(1) - \mathbb{E}_h(1) \right)^\top \left( \text{MaxPool}(\text{ReLU}(\text{BatchNorm}(\text{Conv}(\Gamma(1), x); \alpha(1), 0))) \right) \tag{32}
\end{aligned}$$

In line (a), we approximate  $\mathbb{E}_{h(y,z;0)}$  by its empirical value  $\mathbb{E}_i$ . The de-mean and normalize operators with the scale parameter  $\alpha(1)$  and the shift parameter  $\beta(1) = 0$  in the equations above already have the form of batch normalization. Note that when the model is trained with Stochastic Gradient Descent (SGD), the scale and shift parameters at each layer also account for the error in evaluating statistics of the activations using the mini-batches. Thus,  $\beta(1)$  is not 0 any more, but a parameter learned during the training. Also, in the equations above,  $\psi(1)$  is the activations at layer 1 in CNNs and given by:

$$\psi(1) = \text{MaxPool}(\text{ReLU}(\text{Normalize}(\text{Demean}(\text{Conv}(\Gamma(1), x))))), \quad (33)$$

where  $\sigma_\psi(1)$  is the standard deviation of the  $\psi(1)$ . Eqn.32 can be expressed in term of  $\psi(1)$  as follows:

$$\begin{aligned} & \max_{z(2)} \left( h(1) - \mathbb{E}_h(1) \right)^\top \psi(1) + \text{const} \\ &= \max_{z(2)} h^\top(1)(\psi(1) - \mathbb{E}_\psi(1)) + (h^\top(1)\mathbb{E}_\psi(1) - \mathbb{E}_h^\top(1)\psi(1)) + \text{const} \\ &= \max_{z(2)} \sum_{p_2 \in \mathcal{P}(2)} \left( s(2, p_2) \mu(y, p_2) \underbrace{\frac{\sigma_\psi(2, p_2)}{\sigma_{\mu(y)}(p_2)}}_{\alpha(2, p_2)} \underbrace{\frac{1}{\sigma_\psi(2, p_2)}}_{\text{normalize}} \right. \\ & \quad \times \underbrace{\left( \Gamma^\top(2, p_2) B^\top(2, p_2) T^\top(t; 2, p_2) \psi(1) - \mathbb{E}[\Gamma^\top(2, p_2) B^\top(2, p_2) T^\top(t; 2, p_2) \psi(1)] \right)}_{\text{de-mean}} \\ & \quad \left. + \underbrace{h(1, p_1) \mathbb{E}_\psi(1, p_1) - \mathbb{E}_h(1, p_1) \psi(1, p_1)}_{\beta(2, p_2)} \right) + \text{const} \\ &= \max_{z(2)} \mu^\top(y) \left( \text{MaxPool}(\text{ReLU}(\text{BatchNorm}(\text{Conv}(\Gamma(2), \psi(1)); \alpha(2), \alpha(2) \odot \beta(2)))) \right) + \text{const} \quad (34) \end{aligned}$$

The batch normalization at this layer of the CNN has the scale parameter  $\alpha(2)$  and the shift parameter is the element-wise product of  $\alpha(2)$  and  $\beta(2)$ .

## C.7 Proofs for connection between NRM and cross entropy

**PROOF OF THEOREM B.1** (a) To ease the presentation of proof, we denote the following key notation

$$A := \max_{(z_i)_{i=1}^n, \theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \ln p(y_i | x_i, z_i; \theta).$$



From the definition of  $\mathcal{A}_\gamma = \{\theta : \|h(y, z; 0)\| = \gamma\}$ , we achieve the following equations

$$\begin{aligned}
A &= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log p(y_i | x_i, z_i; \theta) \\
&= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log \frac{p(x_i | y_i, z_i; \theta) p(y_i | z_i; \theta)}{\sum_{y=1}^K p(x_i | y, z_i; \theta) p(y | z_i; \theta)} \\
&= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \left\{ \log p(x_i | y_i, z_i; \theta) + \log p(y_i | z_i; \theta) - \log \left( \sum_{y=1}^K p(x_i | y, z_i; \theta) p(y | z_i; \theta) \right) \right\}.
\end{aligned}$$

From the formulation of NRM, we have the following formulation of prior probabilities  $p(y, z | \theta)$

$$p(y, z | \theta) = \frac{\exp\left(\frac{\eta(y, z)}{\sigma^2}\right) \pi_y}{\sum_{y', z'} \exp\left(\frac{\eta(y', z')}{\sigma^2}\right) \pi_{y'}}.$$

for all  $(y, z)$ . By means of the previous equations, we eventually obtain that

$$\begin{aligned}
A &= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \left\{ \log p(x_i | y_i, z_i; \theta) + \log \left( \frac{\exp\left(\frac{\eta(y_i, z_i)}{\sigma^2}\right) \pi_{y_i}}{\sum_{y, z} \exp\left(\frac{\eta(y, z)}{\sigma^2}\right) \pi_y} \right) \right. \\
&\quad \left. - \log \sum_{y=1}^K \left( p(x_i | y, z_i; \theta) \frac{\exp\left(\frac{\eta(y, z_i)}{\sigma^2}\right) \pi_y}{\sum_{y', z'} \exp\left(\frac{\eta(y', z')}{\sigma^2}\right) \pi_{y'}} \right) \right\} \\
&= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \left( \log p(x_i | y_i, z_i; \theta) + \log \left( \exp\left(\frac{\eta(y_i, z_i)}{\sigma^2}\right) \pi_{y_i} \right) \right) \\
&\quad - \log \left( \sum_{y=1}^K \left( p(x_i | y, z_i; \theta) \exp\left(\frac{\eta(y, z_i)}{\sigma^2}\right) \pi_y \right) \right).
\end{aligned}$$

By defining  $\psi_i(y, z_i) := \log p(x_i | y, z_i; \theta) + \log \left( \exp\left(\frac{\eta(y, z_i)}{\sigma^2}\right) \pi_y \right)$  for all  $1 \leq i \leq n$ , the above equation can be rewritten as

$$\begin{aligned}
A &= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \left( \log (\exp(\psi_i(y_i, z_i))) - \log \left( \sum_{y=1}^K \exp(\psi_i(y, z_i)) \right) \right) \\
&= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log (\text{Softmax}(\psi_i(y_i, z_i))) \\
&\geq \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log \left( \text{Softmax} \left( \max_{z_i} \psi_i(y_i, z_i) \right) \right) = B.
\end{aligned}$$

By means of direct computation, the following equations hold

$$\begin{aligned}
B &= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log \left( \text{Softmax} \left( \max_{z_i} \log p(x_i | y_i, z_i; \theta) + \eta(y_i, z_i) / \sigma^2 + \log \pi_{y_i} \right) \right) \\
&= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log \left( \text{Softmax} \left( \max_{z_i} -\frac{\|x_i - \mu_{y_i, z_i}\|^2}{2\sigma^2} + \frac{\eta(y_i, z_i)}{\sigma^2} + \log \pi_{y_i} \right) \right) \\
&\stackrel{(i)}{=} \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log \left( \text{Softmax} \left( \max_{z_i} \left( \frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2} \right) + b_{y_i} \right) \right) \\
&= -\min_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n -\log \left( \text{Softmax} \left( \max_{z_i} \left( \frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2} \right) + b_{y_i} \right) \right) \\
&= -\min_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n -\log q(y_i | x_i) = -\min_{\theta \in \mathcal{A}_\Gamma} H_{p,q}(y|x)
\end{aligned}$$

where equation (i) is due to the constant norm assumption with rendered images  $h(y, z; 0)$ . Therefore, we achieve the conclusion of part (a) of the theorem.

(b) Regarding the upper bound, from the definition of  $\bar{z}_i$ , we obtain that

$$\begin{aligned}
&\max_{z_i} \log (\text{Softmax}(\psi_i(y_i, z_i))) - \log \left( \text{Softmax} \left( \max_{z_i} \psi_i(y_i, z_i) \right) \right) \\
&= \log (\text{Softmax}(\psi_i(y_i, \bar{z}_i))) - \log \left( \text{Softmax} \left( \max_{z_i} \psi_i(y_i, z_i) \right) \right) \\
&\leq \log \frac{\max_{z_i} \exp(\psi_i(y_i, z_i))}{\sum_{y=1}^K \exp(\psi_i(y, \bar{z}_i))} - \log \left( \text{Softmax} \left( \max_{z_i} \psi_i(y_i, z_i) \right) \right) \\
&= \log \left( \sum_{y=1}^K \exp(\max_{z_i} \psi_i(y, z_i)) \right) - \log \left( \sum_{y=1}^K \exp(\psi_i(y, \bar{z}_i)) \right) \\
&\leq \log K + \max_y \max_{z_i} \psi_i(y, z_i) - \max_y \psi_i(y, \bar{z}_i) \\
&\leq \log K + \max_y \left( \max_{z_i} \psi_i(y, z_i) - \psi_i(y, \bar{z}_i) \right)
\end{aligned}$$

for any  $1 \leq i \leq n$ . As a consequence, we obtain the conclusion of part (b) of the theorem.

**PROOF OF THEOREM B.4** (a) From the definitions of  $(\bar{y}_i, \bar{z}_i)$  and  $(\tilde{y}_i, \tilde{z}_i)$ , we obtain that

$$\begin{aligned}
U_n &= \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{2} \|x_i - h(\bar{y}_i, \bar{z}_i; 0)\|^2 - \log \pi_{\bar{y}_i, \bar{z}_i} \right) \\
&= \min_{\theta} \left\{ \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{2} \|x_i - h(\bar{y}_i, \bar{z}_i; 0)\|^2 - \eta(\bar{y}_i, \bar{z}_i) - \log \pi_{\bar{y}_i} \right) \right. \\
&\quad \left. + \log \left( \sum_{(y', z') \in \mathcal{J}} \exp(\eta(y', z') + \log \pi_{y'}) \right) \right\} = A
\end{aligned}$$

By means of direct computation, the following inequality holds

$$A \leq \min_{\theta} \left\{ \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{2} \|x_i - h(\tilde{y}_i, \tilde{z}_i; 0)\|^2 - \eta(\tilde{y}_i, \tilde{z}_i) - \log \pi_{\tilde{y}_i} \right) + \log \left( \sum_{(y', z') \in \mathcal{J}} \exp(\eta(y', z') + \log \pi_{y'}) \right) \right\}.$$

It is clear that

$$\sum_{(y', z') \in \mathcal{J}} \exp(\eta(y', z') + \log \pi_{y'}) \leq |\mathcal{L}| \sum_{y'} \exp(\max_{z' \in \mathcal{L}} \eta(y', z') + \log \pi_{y'}).$$

Combining this inequality with the inequality of the term  $A$  in the above display, we have

$$\begin{aligned} U_n &\leq \min_{\theta} \left\{ \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{2} \|x_i - h(\tilde{y}_i, \tilde{z}_i; 0)\|^2 - \eta(\tilde{y}_i, \tilde{z}_i) - \log \pi_{\tilde{y}_i} \right) + (\log \pi_{\tilde{y}_i} - \log \pi_{\bar{y}_i}) \right. \\ &\quad \left. + \log \left( \sum_{y'} \exp(\max_{z' \in \mathcal{L}} \eta(y', z') + \log \pi_{y'}) \right) + \log |\mathcal{L}| \right\} \\ &= \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left\{ \left( \frac{\|x_i - h(\tilde{y}_i, \tilde{z}_i; 0)\|^2}{2} - \log(\pi_{\tilde{y}_i, \tilde{z}_i}) \right) + \left( \log \pi_{\tilde{y}_i} - \log \pi_{\bar{y}_i} \right) \right\} + \log |\mathcal{L}| \\ &\leq \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left( \frac{\|x_i - h(\tilde{y}_i, \tilde{z}_i; 0)\|^2}{2} - \log(\pi_{\tilde{y}_i, \tilde{z}_i}) \right) + \log \left( \frac{1}{\bar{\gamma}} - 1 \right) + \log |\mathcal{L}| \\ &= V_n + \log \left( \frac{1}{\bar{\gamma}} - 1 \right) + \log |\mathcal{L}| \end{aligned}$$

where the final inequality is due to the fact that  $\pi_{\tilde{y}_i}/\pi_{\bar{y}_i} \leq (1 - \bar{\gamma})/\bar{\gamma}$  for all  $1 \leq i \leq n$ . Therefore, we achieve the conclusion of part (a) of the theorem.

(b) Similar to the proof argument with part (a), we have

$$\begin{aligned} U_n &= \min_{\theta} \left\{ \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{2} \|x_i - h(\bar{y}_i, \bar{z}_i; 0)\|^2 - \eta(\bar{y}_i, \bar{z}_i) - \log \pi_{\bar{y}_i} \right) + \log \left( \sum_{(y', z') \in \mathcal{J}} \exp(\eta(y', z') + \log \pi_{y'}) \right) \right\} \\ &\geq \min_{\theta} \left\{ \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{2} \|x_i\|^2 - h(\tilde{y}_i, \tilde{z}_i; 0)^{\top} x_i - \eta(\tilde{y}_i, \tilde{z}_i) + \|h(\bar{y}_i, \bar{z}_i; 0)\|^2/2 - \log \pi_{\bar{y}_i} \right) + \log \left( \sum_{y'} \exp(\max_{z' \in \mathcal{L}} \eta(y', z') + \log \pi_{y'}) \right) \right\} = B. \end{aligned}$$

Direct computation with  $B$  leads to

$$\begin{aligned}
B &= \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left\{ \left( \frac{\|x_i - h(\tilde{y}_i, \tilde{z}_i; 0)\|^2}{2} - \log(\pi_{\tilde{y}_i, \tilde{z}_i}) \right) + \left( \log \pi_{\bar{y}_i} - \log \pi_{\tilde{y}_i} \right) \right. \\
&\quad \left. + \|h(\bar{y}_i, \bar{z}_i; 0)\|^2 - \|h(\tilde{y}_i, \tilde{z}_i; 0)\|^2 \right\} \\
&\geq \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left( \frac{\|x_i - h(\tilde{y}_i, \tilde{z}_i; 0)\|^2}{2} - \log(\pi_{\tilde{y}_i, \tilde{z}_i}) \right) + \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left( \log \pi_{\bar{y}_i} - \log \pi_{\tilde{y}_i} \right) \\
&\quad + \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left( \|h(\bar{y}_i, \bar{z}_i; 0)\|^2 - \|h(\tilde{y}_i, \tilde{z}_i; 0)\|^2 \right) \\
&\geq V_n + \log \left( \frac{\bar{\gamma}}{1 - \bar{\gamma}} \right) + \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left( \|h(\bar{y}_i, \bar{z}_i; 0)\|^2 - \|h(\tilde{y}_i, \tilde{z}_i; 0)\|^2 \right)
\end{aligned}$$

where the final inequality is due to the fact that  $\pi_{\tilde{y}_i}/\pi_{\bar{y}_i} \geq \bar{\gamma}/(1 - \bar{\gamma})$  for all  $1 \leq i \leq n$ . As a consequence, we achieve the conclusion of part (b) of the theorem.

## C.8 Proofs for statistical guarantee and generalization bound for (semi)-supervised learning

**PROOF OF THEOREM B.6** The proof of this theorem relies on several results with uniform laws of large numbers. In particular, we will need to demonstrate the following results

$$\begin{aligned}
&\sup_{\theta} \left| \frac{1}{n_1} \sum_{i=1}^{n_1} \max_{(y,z) \in \mathcal{J}} \left( h^{\top}(y, z; 0) x_i + \tau(\theta_{y,z}) \right) \right. \\
&\quad \left. - \int \max_{(y,z) \in \mathcal{J}} \left( h^{\top}(y, z; 0) x + \tau(\theta_{y,z}) \right) dP(x) \right| \rightarrow 0, \tag{35}
\end{aligned}$$

$$\begin{aligned}
&L_{n_1} = \sup_{\theta} \left| \frac{1}{n_1} \sum_{i=1}^{n_1} \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y,z) \in \mathcal{J}} \left( h^{\top}(y, z; 0) x_i + \tau(\theta_{y,z}) \right) \right\} \left( \frac{\|h(y', z'; 0)\|^2}{2} - \log \pi_{y'} \right) \right. \\
&\quad \left. - \int \left( \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y,z) \in \mathcal{J}} \left( h^{\top}(y, z; 0) x + \tau(\theta_{y,z}) \right) \right\} \left( \frac{\|h(y', z'; 0)\|^2}{2} - \log \pi_{y'} \right) \right) dP(x) \right| \rightarrow 0, \tag{36}
\end{aligned}$$

$$\begin{aligned}
&\sup_{\theta} \left| \frac{1}{n - n_1} \sum_{i=n_1+1}^n \max_{z \in \mathcal{L}} \left( h^{\top}(y_i, z; 0) x_i + \tau(\theta_{y_i, z}) \right) \right. \\
&\quad \left. - \int \max_{z \in \mathcal{L}} \left( h^{\top}(y, z; 0) x + \tau(\theta_{y,z}) \right) dQ(x, c) \right| \rightarrow 0, \tag{37}
\end{aligned}$$

$$\begin{aligned}
&E_n^{(1)} = \sup_{\theta} \left| \frac{1}{n - n_1} \sum_{i=n_1+1}^n \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( h^{\top}(y_i, z; 0) x_i + \tau(\theta_{y_i, z}) \right) \right\} \left( \frac{\|h(y_i, z'; 0)\|^2}{2} - \log \pi_{y_i} \right) \right. \\
&\quad \left. - \int \left( \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{g \in \mathcal{J}} \left( h^{\top}(y, z; 0) x + \tau(\theta_{y,z}) \right) \right\} \left( \frac{\|h(y, z'; 0)\|^2}{2} - \log \pi_y \right) \right) dQ(x, c) \right| \rightarrow 0, \tag{38}
\end{aligned}$$

$$E_n^{(2)} = \sup_{\theta} \left| \frac{1}{n - n_1} \sum_{i=n_1+1}^n \log q_{\theta}(y_i|x_i) - \int \log q_{\theta}(y|x) dQ(x, c) \right| \rightarrow 0, \quad (39)$$

$$E_n^{(3)} = \sup_{\theta} \left| \frac{1}{n} \sum_{i=1}^n \sum_{y=1}^K q_{\theta}(y|x_i) \log \left( \frac{q_{\theta}(y|x_i)}{\pi_y} \right) - \int \sum_{y=1}^K q_{\theta}(y|x) dQ(x, c) \log \left( \frac{q_{\theta}(y|x)}{\pi_y} \right) dP(x) \right| \rightarrow 0, \quad (40)$$

almost surely as  $n \rightarrow \infty$ . The proof for (37) is similar to that of (50); therefore, it is omitted.

**Proof of (50):** It is clear that

$$\begin{aligned} \sup_{\theta} \left| \frac{1}{n_1} \sum_{i=1}^{n_1} \max_{(y,z) \in \mathcal{J}} \left( h^{\top}(y, z; 0) x_i + \tau(\theta_{y,z}) \right) - \int \max_{(y,z) \in \mathcal{J}} \left( h^{\top}(y, z; 0) x + \tau(\theta_{y,z}) \right) dP(x) \right| \\ \leq \sup_{|S'| \leq |\mathcal{J}|} \left| \frac{1}{n_1} \sum_{i=1}^{n_1} \max_{s \in S'} s^{\top} [x_i, 1] - \int \max_{s \in S'} s^{\top} [x, 1] dP(x) \right| \end{aligned}$$

where  $[x, 1] \in \mathbb{R}^{D^{(0)}+1}$  denotes the vector forms by concatenating 1 to  $x \in \mathbb{R}^{D^{(0)}}$  and  $S'$  in the above supremum is the set of finite elements in  $\mathbb{R}^{D^{(0)}+1}$ . Therefore, to achieve the result of (50), it is sufficient to show that

$$\sup_{|S'| \leq |\mathcal{J}|} \left| \frac{1}{n_1} \sum_{i=1}^{n_1} \max_{s \in S'} s^{\top} [x_i, 1] - \int \max_{s \in S'} s^{\top} [x, 1] dP(x) \right| \rightarrow 0. \quad (41)$$

To obtain the conclusion of (41), we utilize the classical result with bracketing entropy to establish the uniform laws of large number (cf. Lemma 3.1 in [56]). In particular, we denote  $\mathcal{G}$  to be the family of function on  $\mathbb{R}^{D^{(0)}}$  with the form  $f_{S'}(x) = \max_{s \in S'} s^{\top} [x, 1]$  where  $S' \in \mathcal{O}_{|\mathcal{J}|}$ , which contains all sets that have at most  $|\mathcal{J}|$  elements in  $\mathbb{R}^{D^{(0)}+1}$ . Due to the assumption with distribution  $P$ , we can restrict  $\mathcal{O}_{|\mathcal{J}|}$  to contain only set  $S'$  with elements in  $\mathbb{B}(R)$ , which is a closed ball of radius  $R$  on  $\mathbb{R}^{D^{(0)}+1}$ . By means of Lemma 2.5 in [56], we can find a finite set  $E_{\delta}$  such that each element in  $\mathbb{B}(R)$  is within distance  $\delta$  to some element of  $E_{\delta}$  for all  $\delta > 0$ . We denote  $\mathcal{O}_{|\mathcal{J}|}(\delta)$  to be the subset of  $\mathcal{O}_{|\mathcal{J}|}$  such that it only contains sets with elements in  $E_{\delta}$  for all  $\delta > 0$ . Now, for each set  $S' = \{s_1, \dots, s_k\} \in \mathcal{O}_{|\mathcal{J}|}$ , we can choose corresponding set  $\bar{S} = \{s'_1, \dots, s'_k\} \in \mathcal{O}_{|\mathcal{J}|}(\delta)$  such that  $\|s_i - s'_i\| \leq \delta$  for all  $1 \leq i \leq k$ . Now, we denote

$$\begin{aligned} \bar{f}_{\bar{S}}(x) &= \max_{s \in \bar{S}} s^{\top} [x, 1] + \delta \| [x, 1] \|, \\ \underline{f}_{\bar{S}}(x) &= \max_{s \in \bar{S}} s^{\top} [x, 1] - \delta \| [x, 1] \| \end{aligned}$$

for any  $\bar{S} \in \mathcal{O}_{|\mathcal{J}|}(\delta)$ . It is clear that  $\underline{f}_{\bar{S}}(x) \leq f_{S'}(x) \leq \bar{f}_{\bar{S}}(x)$  for all  $x \in \mathbb{R}^{D^{(0)}}$ . Furthermore, we also have that

$$\int (\bar{f}_{\bar{S}}(x) - \underline{f}_{\bar{S}}(x)) dP(x) = 2\delta \int \| [x, 1] \| dP(x) \leq 2\delta \left( \int \|x\| dP(x) + 1 \right).$$

For any  $\epsilon > 0$ , by choosing  $\delta < \frac{\epsilon}{2(\int \|x\| dP(x) + 1)}$  then we will have that  $\int (\bar{f}_{\bar{S}}(x) - \underline{f}_{\bar{S}}(x)) dP(x) < \epsilon$ .

It implies that the  $\epsilon$ -bracketing entropy of  $\mathcal{G}$  is finite for the  $\mathbb{L}_1$  norm with distribution  $P$  (for the definition of bracketing entropy, see Definition 2.2 in [56]). According to Lemma 3.1 in [56], it implies that  $\mathcal{G}$  satisfies the uniform law of large numbers, i.e., (41) holds.

**Proof of (51):** To achieve the conclusion of this claim, we will need to rely on the control of Rademacher complexity based on Vapnik-Chervonenkis (VC) dimension. In particular, we firstly demonstrate that

$$\sup_{|S'|=k} \left| \frac{1}{n_1} \sum_{i=1}^{n_1} 1_{\left\{j=\arg \max_{1 \leq l \leq |S'|} [x_i, 1]^\top s'_l\right\}} - \int 1_{\left\{j=\arg \max_{1 \leq l \leq |S'|} [x, 1]^\top s'_l\right\}} dP(x) \right| \rightarrow 0 \quad (42)$$

almost surely as  $n_1 \rightarrow \infty$  for each  $1 \leq j \leq k$  and  $k \geq 1$  where the supremum is taken with respect to  $S' = \{s'_1, \dots, s'_k\}$ . For each  $j$ , we denote the Rademacher complexity as follows

$$R_{n_1} = \mathbb{E} \sup_{|S'|=k} \left| \frac{1}{n_1} \sum_{i=1}^{n_1} \sigma_i 1_{\left\{j=\arg \max_{1 \leq l \leq |S'|} [x_i, 1]^\top s'_l\right\}} \right|$$

where  $\sigma_1, \dots, \sigma_{n_1}$  are i.i.d. Rademacher random variables, i.e.,  $\mathbb{P}(\sigma_i = -1) = \mathbb{P}(\sigma_i = 1) = 1/2$  for  $1 \leq i \leq n_1$ . Then, for any  $n_1 \geq 1$  and  $\delta \geq 0$ , according to standard argument with Rademacher complexity [57],

$$\sup_{|S'|=k} \left| \frac{1}{n_1} \sum_{i=1}^{n_1} 1_{\left\{j=\arg \max_{1 \leq l \leq |S'|} [x_i, 1]^\top s'_l\right\}} - \int 1_{\left\{j=\arg \max_{1 \leq l \leq |S'|} [x, 1]^\top s'_l\right\}} dP(x) \right| \leq 2R_{n_1} + \delta$$

with probability at least  $1 - 2 \exp\left(-\frac{n_1 \delta^2}{8}\right)$ . According to Borel-Cantelli's lemma, to achieve (42), it is sufficient to demonstrate that  $R_{n_1} \rightarrow 0$  as  $n_1 \rightarrow \infty$ .

To achieve that result, we will utilize the study of VC dimension with partitions (cf. Section 21.5 in [58]). In particular, for each set  $S' = (s'_1, \dots, s'_k)$ , it gives rise to the partition  $A_i = \{x \in \mathbb{R}^{D(0)} : [x, 1]^\top s'_i \geq [x, 1]^\top s'_l \forall l \in \{1, \dots, k\}\}$  as  $1 \leq i \leq k$ . For our purpose with (42), it is sufficient to consider  $\mathcal{P}_{n_1} = \left\{A_j, \bigcup_{i \neq j} A_i\right\}$ , which is a partition of  $\mathbb{B}(R)$ , for each set  $S'$  with  $k$  elements. We denote  $\mathcal{F}$  to be the collection of all  $\mathcal{P}_n$  for all  $S'$  with  $k$  elements and  $\mathcal{B}(\mathcal{P}_n)$  the collection of all sets obtained from the unions of elements of  $\mathcal{P}_n$ . For each data  $(x_1, \dots, x_{n_1})$ , we let  $N_{\mathcal{F}}(x_1, \dots, x_{n_1})$  the number of different sets in  $\{(x_1, \dots, x_{n_1}) \cap A : A \in \mathcal{B}(\mathcal{P}_{n_1}) \text{ for } \mathcal{P}_{n_1} \in \mathcal{F}\}$ . The shatter coefficient of  $\mathcal{F}$  is defined as

$$\Delta_{n_1}(\mathcal{F}) = s(\mathcal{F}, n_1) = \max_{(x_1, \dots, x_{n_1})} N_{\mathcal{F}}(x_1, \dots, x_{n_1}).$$

According to Lemma 21.1 in [58],  $\Delta_{n_1}(\mathcal{F}) \leq 4\Delta_{n_1}^*(\mathcal{F})$  where  $\Delta_{n_1}^*(\mathcal{F})$  is the maximal number of different ways that  $n_1$  points can be partitioned by members of  $\mathcal{F}$ . Now, for each element

$\mathcal{P}_{n_1} = \left\{A_j, \bigcup_{i \neq j} A_i\right\}$  of  $\mathcal{F}$ , it is clear that the boundaries between  $A_j$  and  $\bigcup_{i \neq j} A_i$  are subsets of

hyperplanes. From the formulation of  $A_j$ , we have at most  $k - 1$  boundaries between  $A_j$  and  $\bigcup_{i \neq j} A_i$ .

From the classical result of [59], each  $n_1$  points in  $\mathbb{B}(R)$  can be splitted by a hyperplane in at most  $n_1^{D^{(0)}+1}$  different ways as the VC dimension of the hyperplane is at most  $D^{(0)} + 1$ . As a consequence, we would have  $\Delta_{n_1}^*(\mathcal{F}) \leq n_1^{(D^{(0)}+1)(k-1)}$ , which leads to  $\Delta_{n_1}(\mathcal{F}) \leq 4n_1^{(D^{(0)}+1)(k-1)}$ .

Going back to our evaluation with Rademacher complexity  $R_{n_1}$ , by means of Massart's lemma, we have that

$$\begin{aligned} R_{n_1} &= \mathbb{E} \left( \mathbb{E}_\sigma \sup_{|S'|=k} \left| \frac{1}{n_1} \sum_{i=1}^{n_1} \sigma_i 1_{\left\{ j = \arg \max_{1 \leq l \leq |S'|} [x_i, 1]^\top s'_l \right\}} \right| \middle| x_1, \dots, x_{n_1} \right) \\ &\leq \mathbb{E} \left( \sqrt{\frac{2 \log 2N_{\mathcal{F}}(x_1, \dots, x_{n_1})}{n_1}} \right) \leq \sqrt{\frac{2(\log 8 + (D^{(0)} + 1)(k - 1) \log n_1)}{n_1}} \rightarrow 0 \end{aligned} \quad (43)$$

as  $n_1 \rightarrow \infty$ . Therefore, (42) is proved.

**Proof of (38):** To achieve the conclusion of this claim, we firstly demonstrate that

$$\begin{aligned} \sup_{|S'|=k} \left| \frac{1}{n - n_1} \sum_{i=n_1+1}^n 1_{\left\{ j = \arg \max_{1 \leq l \leq |S'|} [x_i, 1]^\top s'_l \right\}} 1_{\{y_i=l\}} \right. \\ \left. - \int 1_{\left\{ j = \arg \max_{1 \leq l \leq |S'|} [x, 1]^\top s'_l \right\}} 1_{\{c=l\}} dQ(x, c) \right| \rightarrow 0 \end{aligned} \quad (44)$$

almost surely as  $n \rightarrow \infty$  for each  $1 \leq j \leq k$  and  $1 \leq l \leq K$  where  $k \geq 1$  and the supremum is taken with respect to  $S' = \{s'_1, \dots, s'_k\}$ . The proof of the above result will rely on VC dimension with Voronoi partitions being established in (51). In particular, according to the standard argument with Rademacher complexity, it is sufficient to demonstrate that

$$R'_n = \mathbb{E} \sup_{|S'|=k} \left| \frac{1}{n - n_1} \sum_{i=n_1+1}^n \sigma_i 1_{\left\{ j = \arg \max_{1 \leq l \leq |S'|} [x_i, 1]^\top s'_l \right\}} 1_{\{y_i=l\}} \right| \rightarrow 0.$$

By means of the inequality with Rademacher complexity in (43), we obtain that

$$\begin{aligned} R'_n &= \sum_{v=0}^{n-n_1} \sum_{\vec{c} \in A_v} \mathbb{E} \left( \mathbb{E}_\sigma \sup_{|S'|=k} \left| \frac{1}{n - n_1} \sum_{i=n_1+1}^n \sigma_i 1_{\left\{ j = \arg \max_{1 \leq l \leq |S'|} [x_i, 1]^\top s'_l \right\}} 1_{\{y_i=l\}} \right| \middle| \vec{c} \in A_v \right) \mathbb{P}(\vec{c} \in A_v) \\ &= \sum_{v=0}^{n-n_1} \mathbb{E} \sup_{|S'|=k} \left| \frac{1}{n - n_1} \sum_{i=n_1+1}^{n_1+v} \sigma_i 1_{\left\{ j = \arg \max_{1 \leq l \leq |S'|} [x_i, 1]^\top s'_l \right\}} \right| p_l^v (1 - p_l)^{n-n_1-v} \binom{n - n_1}{v} \\ &\leq \sum_{v=1}^{n-n_1} \frac{v}{n - n_1} \sqrt{\frac{2(\log 8 + (D^{(0)} + 1)(k - 1) \log v)}{v}} p_l^v (1 - p_l)^{n-n_1-v} \binom{n - n_1}{v} \end{aligned}$$

$$\begin{aligned}
&\leq \sqrt{\frac{2\log 8}{n-n_1}} \sum_{v=1}^{n-n_1} \sqrt{\frac{v}{n-n_1}} p_l^v (1-p_l)^{n-n_1-v} \binom{n-n_1}{v} \\
&+ \sqrt{\frac{2(D^{(0)}+1)(k-1)\log(n-n_1)}{n-n_1}} \sum_{v=1}^{n-n_1} \sqrt{\frac{v \log v}{(n-n_1)\log(n-n_1)}} p_l^v (1-p_l)^{n-n_1-v} \binom{n-n_1}{v} \\
&\leq \left( \sqrt{\frac{2\log 8}{n-n_1}} + \sqrt{\frac{2(D^{(0)}+1)(k-1)\log(n-n_1)}{n-n_1}} \right) \sum_{v=1}^{n-n_1} p_l^v (1-p_l)^{n-n_1-v} \binom{n-n_1}{v} \\
&= \left( \sqrt{\frac{2\log 8}{n-n_1}} + \sqrt{\frac{2(D^{(0)}+1)(k-1)\log(n-n_1)}{n-n_1}} \right) (1 - (1-p_l)^{n-n_1}) \rightarrow 0
\end{aligned}$$

as  $n \rightarrow \infty$  where  $\vec{c} = (c_{n-n_1+1}, \dots, c_n)$  and  $A_v$  is the set of  $\vec{c}$  such that there are exactly  $v$  values of  $y_i$  to be  $l$  for  $0 \leq v \leq n - n_1$ . The final inequality is due to the fact that  $v/(n-1) \leq 1$  and  $v \log v / \{(n-1) \log(n-1)\} \leq 1$  for all  $1 \leq v \leq n - n_1$ . Therefore, we achieve the conclusion of (44).

Now, coming back to (38), by means of triangle inequality, we achieve that

$$\begin{aligned}
E_n^{(1)} &\leq \sum_{l=1}^K \sup_{\theta} \left| \frac{1}{n-n_1} \sum_{i=n_1+1}^n \sum_{z' \in \mathcal{L}} 1_{\left\{ z' = \arg \max_{z \in \mathcal{L}} \left( h^\top(y_i, z; 0) x_i + \tau(\theta_{y_i, z}) \right) \right\}} 1_{\{y_i=l\}} \left( \frac{\|h(y_i, z'; 0)\|^2}{2} - \log \pi_{y_i} \right) \right. \\
&\quad \left. - \int \left( \sum_{z' \in \mathcal{L}} 1_{\left\{ z' = \arg \max_{g \in \mathcal{J}} \left( h^\top(y, z; 0) x + \tau(\theta_{y, z}) \right) \right\}} 1_{\{y=l\}} \left( \frac{\|h(y, z'; 0)\|^2}{2} - \log \pi_y \right) \right) dQ(x, y) \right| \\
&\leq \sum_{l=1}^K \sup_{\theta} \sum_{z' \in \mathcal{L}} \left| \frac{\|h(l, z'; 0)\|^2}{2} - \log \pi_l \right| \left| \frac{1}{n-n_1} \sum_{i=n_1+1}^n 1_{\left\{ z' = \arg \max_{z \in \mathcal{L}} \left( h^\top(y_i, z; 0) x_i + \tau(\theta_{y_i, z}) \right) \right\}} 1_{\{y_i=l\}} \right. \\
&\quad \left. - \int 1_{\left\{ z' = \arg \max_{g \in \mathcal{J}} \left( h^\top(y, z; 0) x + \tau(\theta_{y, z}) \right) \right\}} 1_{\{y=l\}} dQ(x, y) \right| \rightarrow 0
\end{aligned}$$

where the last inequality is due to the results with uniform laws of large numbers from (44) and the fact that  $\frac{\|h(l, z'; 0)\|^2}{2} - \log \pi_l$  is bounded for all  $l$  and  $z' \in \mathcal{L}$ . Hence, we obtain the conclusion of (38).

**Proof of (39):** For this claim, we have the following inequality

$$\begin{aligned}
E_n^{(2)} &\leq \sup_{\{S'_y\}, \{\pi_y\}} \left| \frac{1}{n-n_1} \sum_{i=n_1+1}^n \log \frac{\exp \left( \max_{s \in S'_{y_i}} s^\top [x_i, 1] + \log \pi_{y_i} \right)}{\sum_{\ell=1}^K \exp \left( \max_{s \in S'_l} s^\top [x_i, 1] + \log \pi_l \right)} \right. \\
&\quad \left. - \int \log \frac{\exp \left( \max_{s \in S'_y} s^\top [x, 1] + \log \pi_y \right)}{\sum_{\ell=1}^K \exp \left( \max_{s \in S'_l} s^\top [x, 1] + \log \pi_l \right)} dQ(x, y) \right|
\end{aligned}$$



$$\begin{aligned} &\leq \sum_{\ell=1}^K \sup_{\{S'_y\}, \{\pi_y\}} \left| \frac{1}{n - n_1} \sum_{i=n_1+1}^n \log \frac{\exp\left(\max_{s \in S'_{y_i}} s^\top [x_i, 1] + \log \pi_{y_i}\right)}{\sum_{\ell=1}^K \exp\left(\max_{s \in S'_\ell} s^\top [x_i, 1] + \log \pi_\ell\right)} 1_{\{y_i=l\}} \right. \\ &\quad \left. - \int \log \frac{\exp\left(\max_{s \in S'_y} s^\top [x, 1] + \log \pi_y\right)}{\sum_{\ell=1}^K \exp\left(\max_{s \in S'_\ell} s^\top [x, 1] + \log \pi_\ell\right)} 1_{\{c=l\}} dQ(x, y) \right| = \sum_{\ell=1}^K F_{n,\ell} \end{aligned}$$

where  $\{S'_y\}$  in the above supremum stands for the collection of sets  $S'_1, \dots, S'_K$  such that  $|S'_y| \leq |\mathcal{L}|$  and elements in  $S'_y$  are in  $\mathbb{R}^{D^{(0)}+1}$ . Therefore, to achieve the conclusion of (39), it is sufficient to demonstrate that  $F_{n,\ell} \rightarrow 0$  almost surely as  $n \rightarrow \infty$  for each  $1 \leq \ell \leq K$ .

In fact, for each  $1 \leq \ell \leq K$ , we denote  $\mathcal{G}$  to be the family of function on  $\mathbb{R}^{D^{(0)}} \times \{1, \dots, K\}$  of the form

$$f_{\{S'_y\}, \{\pi_y\}}(x, y) = \log \frac{\exp\left(\max_{s \in S'_y} s^\top [x, 1] + \log \pi_y\right)}{\sum_{\ell=1}^K \exp\left(\max_{s \in S'_\ell} s^\top [x, 1] + \log \pi_\ell\right)} 1_{\{y=l\}}$$

for all  $(x, y)$  where  $S'_1, \dots, S'_K \in \mathcal{O}_{|\mathcal{L}|}$ , which contains all sets that have at most  $|\mathcal{L}|$  elements in  $\mathbb{R}^{D^{(0)}+1}$ , and  $\{\pi_y\}$  satisfy that  $\sum_{y=1}^K \pi_y = 1$  and  $\pi_y \geq \bar{\gamma}$  for all  $1 \leq y \leq K$ . Due to the assumption with distribution  $P$ , we can restrict  $\mathcal{O}_{|\mathcal{L}|}$  to contain only set  $S'$  with elements in the ball  $\mathbb{B}(R)$  of radius  $R$  on  $\mathbb{R}^{D^{(0)}+1}$ . By means of Lemma 2.5 in [56], we can find a finite set  $E_\delta$  such that each element in  $\mathbb{B}(R)$  is within distance  $\delta$  to some element of  $E_\delta$  for all  $\delta > 0$ . Additionally, there exists a set  $\Delta(\delta)$  such that for each  $(\pi_1, \dots, \pi_K)$ , we can find a corresponding element  $(\pi'_1, \dots, \pi'_K) \in \Delta(\delta)$  such that  $\|(\pi_1, \dots, \pi_K) - (\pi'_1, \dots, \pi'_K)\| \leq \delta$ . We denote

$$\mathcal{F}(\delta) = \left\{ \{S'_y\}, \{\pi_y\} : \text{elements of } S'_y \text{ in } E_\delta, \text{ and } (\pi_1, \dots, \pi_K) \in \Delta(\delta) \right\}$$

for all  $\delta > 0$ .

For each element  $\{S'_y\}_{y=1}^K, \{\pi_y\}_{y=1}^K$ , we can choose the corresponding element  $\{\bar{S}'_y\}_{y=1}^K, \{\bar{\pi}_y\}_{y=1}^K \in \mathcal{F}(\delta)$  such that  $S'_y = \{s_{y1}, \dots, s_{yk_y}\}$ ,  $\bar{S}'_y = \{s'_{y1}, \dots, s'_{yk_y}\}$  satisfy  $\|s_{yj} - s'_{yj}\| \leq \delta$  for all  $1 \leq y \leq K$  and  $1 \leq j \leq k_y$ . Additionally,  $\|(\pi_1, \dots, \pi_K) - (\bar{\pi}_1, \dots, \bar{\pi}_K)\| \leq \delta$ . With these notations, we define

$$\begin{aligned} \bar{f}_{\{\bar{S}'_y\}, \{\bar{\pi}_y\}}(x, y) &= \log \left( \frac{\exp(\max_{s \in \bar{S}'_y} s^\top [x, 1] + \log \bar{\pi}_y)}{\sum_{\tau=1}^K \exp(\max_{s \in \bar{S}'_\tau} s^\top [x, 1] + \log \bar{\pi}_\tau)} \right) 1_{\{y=l\}} + 2\delta\|[x, 1]\| + 2\delta/\bar{\gamma}, \\ f_{\{\bar{S}'_y\}, \{\bar{\pi}_y\}}(x, y) &= \log \left( \frac{\exp(\max_{s \in \bar{S}'_y} s^\top [x, 1] + \log \bar{\pi}_y)}{\sum_{\tau=1}^K \exp(\max_{s \in \bar{S}'_\tau} s^\top [x, 1] + \log \bar{\pi}_\tau)} \right) 1_{\{y=l\}} - 2\delta\|[x, 1]\| - 2\delta/\bar{\gamma} \end{aligned}$$

for any  $(\{\bar{S}'_y\}, \{\bar{\pi}_y\}) \in \mathcal{F}(\delta)$ . By means of Cauchy-Schwarz's inequality, we have

$$\begin{aligned} s_{yi}^\top [x, 1] 1_{\{y=l\}} - (s'_{yi})^\top [x, 1] 1_{\{y=l\}} &\leq \|s_{ci} - s'_{yi}\| \| [x, 1] \| 1_{\{y=l\}} \\ &\leq \delta \| [x, 1] \| \end{aligned}$$

for all  $x$  and  $1 \leq i \leq k$ . Additionally, the following also holds

$$s_{yi}^\top [x, 1] 1_{\{y=l\}} - (s'_{yi})^\top [x, 1] 1_{\{y=l\}} \geq -\delta \| [x, 1] \|.$$

Furthermore,  $|\log \pi_y - \log \bar{\pi}_y| \leq \log(1 + \delta/\bar{\gamma}) \leq \delta/\bar{\gamma}$ . Hence, we obtain that

$$\begin{aligned} \frac{\exp(\max_{s \in S'_y} s^\top [x, 1] + \log \pi_y)}{\sum_{\tau=1}^K \exp(\max_{s \in S'_\tau} s^\top [x, 1] + \log \pi_\tau)} &\leq \frac{\exp(\max_{s \in \bar{S}'_y} s^\top [x, 1] + \log \bar{\pi}_y + \delta \| [x, 1] \| + \delta/\bar{\gamma})}{\sum_{\tau=1}^K \exp(\max_{s \in \bar{S}'_\tau} s^\top [x, 1] + \log \bar{\pi}_\tau - \delta \| [x, 1] \| - \delta/\bar{\gamma})} \\ &\leq \frac{\exp(\max_{s \in \bar{S}'_y} s^\top [x, 1] + \log \bar{\pi}_y)}{\sum_{\tau=1}^K \exp(\max_{s \in \bar{S}'_\tau} s^\top [x, 1] + \log \bar{\pi}_\tau)} \exp(2\delta \| [x, 1] \| + 2\delta/\bar{\gamma}). \end{aligned}$$

Similarly, we also have

$$\frac{\exp(\max_{s \in S'_y} s^\top [x, 1] + \log \pi_y)}{\sum_{\tau=1}^K \exp(\max_{s \in S'_\tau} s^\top [x, 1] + \log \pi_\tau)} \geq \frac{\exp(\max_{s \in \bar{S}'_y} s^\top [x, 1] + \log \bar{\pi}_y)}{\sum_{\tau=1}^K \exp(\max_{s \in \bar{S}'_\tau} s^\top [x, 1] + \log \bar{\pi}_\tau)} \exp(-2\delta \| [x, 1] \| - 2\delta/\bar{\gamma}).$$

As a consequence, we achieve that

$$\underline{f}_{\{\bar{S}'_y\}, \{\bar{\pi}_y\}}(x, y) \leq f_{\{S'_y\}, \{\pi_y\}}(x, y) \leq \bar{f}_{\{\bar{S}'_y\}, \{\bar{\pi}_y\}}(x, y)$$

for all  $(x, y)$ . With the formulations of  $\underline{f}_{\{\bar{S}'_y\}, \{\bar{\pi}_y\}}(x, c)$  and  $\bar{f}_{\{\bar{S}'_y\}, \{\bar{\pi}_y\}}(x, y)$ , we have

$$\begin{aligned} &\int \left( \bar{f}_{\{\bar{S}'_y\}, \{\bar{\pi}_y\}}(x, y) - \underline{f}_{\{\bar{S}'_y\}, \{\bar{\pi}_y\}}(x, y) \right) dQ(x, y) \\ &= 4\delta \int \| [x, 1] \| dQ(x, y) + 4\delta/\bar{\gamma} \leq 4\delta \left( \int \| x \| dQ(x, y) + 1 \right) + 4\delta/\bar{\gamma}. \end{aligned}$$

For any  $\epsilon > 0$ , by choosing  $\delta < \frac{\epsilon}{4 \int \| x \| dQ(x, y) + 4 + 4/\bar{\gamma}}$  then we will have  $\int (\bar{f}_{\{\bar{S}'_y\}, \{\bar{\pi}_y\}}(x, y) - \underline{f}_{\{\bar{S}'_y\}, \{\bar{\pi}_y\}}(x, y)) dQ(x, y) < \epsilon$ . It implies that the  $\epsilon$ -bracketing entropy of  $\mathcal{G}$  is finite for the  $L_1$  norm with distribution  $Q$ . Therefore, it implies that  $\mathcal{G}$  satisfies the uniform law of large numbers, i.e.,  $F_{n,l} \rightarrow 0$  almost surely as  $n \rightarrow \infty$  for all  $1 \leq l \leq K$ . As a consequence, the uniform law of large number result (39) holds.

Going back to the original problem, denote  $\tilde{\theta}^0 = \left( \{\tilde{\mu}^0(y)\}_{y=1}^K, \{\tilde{\Gamma}_0(\ell)\}_{\ell=1}^L, \{\tilde{\pi}_y^0\}_{y=1}^K, \{\tilde{b}_0(\ell)\}_{\ell=1}^L \right)$  the optimal solutions of population partially labeled LDCE (Note that, the existence of these optimal

solutions is guaranteed due to the compact assumptions with the parameter spaces  $\Theta_\ell$ ,  $\Omega$ , and  $\Xi_\ell$  for all  $1 \leq \ell \leq L$ ). Then, according to the formulation of partially labeled LDCE, we will have that

$$\begin{aligned}
Y_n \leq & \min_{\theta} \frac{\alpha_{\text{RC}}}{n} \left\{ \sum_{i=1}^{n_1} \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x_i^\top \tilde{h}^0(y, z; 0) + \tau(\tilde{\theta}_{y, z}^0) \right) \right\} \left( \frac{\|x_i - \tilde{h}^0(y', z'; 0)\|^2}{2} \right. \right. \\
& \left. \left. - \log(\tilde{p}^0(y', z')) \right) + \sum_{i=n_1+1}^n \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( x_i^\top \tilde{h}^0(y, z; 0) + \tau(\tilde{\theta}_{y, z}^0) \right) \right\} \left( \frac{\|x_i - \tilde{h}^0(y_i, z'; 0)\|^2}{2} \right. \right. \\
& \left. \left. - \log(\tilde{p}^0(y_i, z')) \right) \right\} - \frac{\alpha_{\text{CE}}}{n - n_1} \sum_{i=n_1+1}^n \log q_{\tilde{\theta}^0}(y_i | x_i) + \frac{\alpha_{\text{KL}}}{n} \sum_{i=1}^n \sum_{y=1}^K q_{\tilde{\theta}^0}(y | x_i) \log \left( \frac{q_{\tilde{\theta}^0}(y | x_i)}{\pi_y} \right) = D_n
\end{aligned}$$

for all  $n \geq 1$  where we have the following formulations

$$\begin{aligned}
\tilde{h}^0(y, z; 0) &= \tilde{\Lambda}_0(z; 1) \dots \tilde{\Lambda}_0(z; L) \tilde{\mu}^0(y), \\
\tilde{\Lambda}_0(z; \ell) &= \sum_{p \in \mathcal{P}(\ell)} s(\ell, p) T(t; \ell, p) B(\ell, p) \tilde{\Gamma}_0(\ell, p), \\
\tilde{p}^0(y', z') &= \exp \left( \tau(\tilde{\theta}_{y', z'}^0) + \log \tilde{\pi}_{y'}^0 \right) / \left( \sum_{y=1}^K \exp \left( \max_{z \in \mathcal{L}} \tau(\tilde{\theta}_{y, z}^0) + \log \tilde{\pi}_y^0 \right) \right).
\end{aligned}$$

From the results with uniform laws of large numbers in (50), (51), (37), (38), (39), and (40), we obtain that

$$\begin{aligned}
& \frac{1}{n_1} \sum_{i=1}^{n_1} \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x_i^\top \tilde{h}^0(y, z; 0) + \tau(\tilde{\theta}_{y, z}^0) \right) \right\} \left( \frac{\|x_i - \tilde{h}^0(y', z'; 0)\|^2}{2} - \log(\tilde{p}^0(y', z')) \right) \\
& \rightarrow \int \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x^\top \tilde{h}^0(y, z; 0) + \tau(\tilde{\theta}_{y, z}^0) \right) \right\} \left( \frac{\|x - \tilde{h}^0(y', z'; 0)\|^2}{2} - \log(\tilde{p}^0(y', z')) \right) dP(x), \\
& \frac{1}{n - n_1} \sum_{i=n_1+1}^n \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( x_i^\top \tilde{h}^0(y, z; 0) + \tau(\tilde{\theta}_{y, z}^0) \right) \right\} \left( \frac{\|x_i - \tilde{h}^0(y_i, z'; 0)\|^2}{2} - \log(\tilde{p}^0(y_i, z')) \right) \\
& \rightarrow \int \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( x^\top \tilde{h}^0(y, z; 0) + \tau(\tilde{\theta}_{y, z}^0) \right) \right\} \left( \frac{\|x - \tilde{h}^0(y, z'; 0)\|^2}{2} - \log(\tilde{p}^0(y, z')) \right) dQ(x, y), \\
& \frac{1}{n - n_1} \sum_{i=n_1+1}^n \log q_{\tilde{\theta}^0}(y_i | x_i) \rightarrow \int \log q_{\tilde{\theta}^0}(y | x) dQ(x, y), \\
& \frac{1}{n} \sum_{i=1}^n \sum_{y=1}^K q_{\tilde{\theta}^0}(y | x_i) \log \left( \frac{q_{\tilde{\theta}^0}(y | x_i)}{\pi_y} \right) \rightarrow \int \sum_{y=1}^K q_{\tilde{\theta}^0}(y | x) \log \left( \frac{q_{\tilde{\theta}^0}(y | x)}{\pi_y} \right) dP(x)
\end{aligned}$$

almost surely as  $n \rightarrow \infty$ . Combining with the fact that  $n_1/n \rightarrow \bar{\lambda}$ , the above results lead to  $D_n \rightarrow \bar{Y}$  almost surely as  $n \rightarrow \infty$ . Therefore, we have  $\lim_{n \rightarrow \infty} Y_n \leq \bar{Y}$  almost surely as  $n \rightarrow \infty$ . On the other

hand, with the results of uniform laws of large numbers in (50), (51), (37), (38), (39), and (40), we also have that

$$\begin{aligned}
& \frac{1}{n_1} \sum_{i=1}^{n_1} \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x_i^\top \tilde{h}(y, z; 0) + \tau(\tilde{\theta}_{y, z}) \right) \right\} \left( \frac{\|x_i - \tilde{h}(y', z'; 0)\|^2}{2} - \log(\tilde{p}(y', z')) \right) \\
& \rightarrow \int \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x^\top \tilde{h}(y, z; 0) + \tau(\tilde{\theta}_{y, z}) \right) \right\} \left( \frac{\|x - \tilde{h}(y', z'; 0)\|^2}{2} - \log(\tilde{p}(y', z')) \right) dP(x), \\
& \frac{1}{n - n_1} \sum_{i=n_1+1}^n \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( x_i^\top \tilde{h}(y, z; 0) + \tau(\tilde{\theta}_{y, z}) \right) \right\} \left( \frac{\|x_i - \tilde{h}(y_i, z'; 0)\|^2}{2} - \log(\tilde{p}(y_i, z')) \right) \\
& \rightarrow \int \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( x^\top \tilde{h}^0(y, z; 0) + \tau(\tilde{\theta}_{y, z}) \right) \right\} \left( \frac{\|x - \tilde{h}(y, z'; 0)\|^2}{2} - \log(\tilde{p}(y, z')) \right) dQ(x, y), \\
& \frac{1}{n - n_1} \sum_{i=n_1+1}^n \log q_{\tilde{\theta}}(y_i | x_i) \rightarrow \int \log q_{\tilde{\theta}}(y | x) dQ(x, y), \\
& \frac{1}{n} \sum_{i=1}^n \sum_{y=1}^K q_{\tilde{\theta}}(y | x_i) \log \left( \frac{q_{\tilde{\theta}}(y | x_i)}{\pi_y} \right) \rightarrow \int \sum_{y=1}^K q_{\tilde{\theta}}(y | x) \log \left( \frac{q_{\tilde{\theta}}(y | x)}{\pi_y} \right) dP(x)
\end{aligned}$$

almost surely as  $n \rightarrow \infty$  where  $\tilde{\theta} = \left( \{\tilde{\mu}(y)\}_{y=1}^K, \{\tilde{\Gamma}(\ell)\}_{\ell=1}^L, \{\tilde{\pi}_y\}_{y=1}^K, \{\tilde{b}(\ell)\}_{\ell=1}^L \right)$  is the optimal solution of partially labeled LDCE and

$$\begin{aligned}
\tilde{h}(y, z; 0) &= \tilde{\Lambda}(z; 1) \dots \tilde{\Lambda}(z; L) \tilde{\mu}(y), \\
\tilde{\Lambda}(z; \ell) &= \sum_{p \in \mathcal{P}(\ell)} s(\ell, p) T(t; \ell, p) B(\ell, p) \tilde{\Gamma}(\ell, p), \\
\tilde{p}(y', z') &= \exp \left( \tau(\tilde{\theta}_{y', z'}) + \log \tilde{\pi}_{y'} \right) / \left( \sum_{y=1}^K \exp \left( \max_{z \in \mathcal{L}} \tau(\tilde{\theta}_{y', z'}) + \log \tilde{\pi}_y \right) \right).
\end{aligned}$$

Hence, we eventually achieve that

$$\begin{aligned}
Y_n & \rightarrow \alpha_{\text{RC}} \left\{ \bar{\lambda} \left( \int \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x^\top \tilde{h}(y, z; 0) + \tau(\tilde{\theta}_{y, z}) \right) \right\} \left( \frac{\|x - \tilde{h}(y', z'; 0)\|^2}{2} \right. \right. \right. \\
& \left. \left. - \log(\tilde{p}(y', z')) \right) dP(x) \right) + (1 - \bar{\lambda}) \left( \int \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( x^\top \tilde{h}^0(y, z; 0) + \tau(\tilde{\theta}_{y, z}) \right) \right\} \left( \frac{\|x - \tilde{h}(y, z'; 0)\|^2}{2} \right. \right. \right. \\
& \left. \left. - \log(\tilde{p}(y, z')) \right) dQ(x, y) \right) \right\} - \alpha_{\text{CE}} \int \log q_{\tilde{\theta}}(y | x) dQ(x, y) \\
& \quad + \alpha_{\text{KL}} \int \sum_{y=1}^K q_{\tilde{\theta}}(y | x) \log \left( \frac{q_{\tilde{\theta}}(y | x)}{\pi_y} \right) dP(x) \geq \bar{Y}
\end{aligned}$$

almost surely as  $n \rightarrow \infty$ . As a consequence, we achieve  $Y_n \rightarrow \bar{Y}$  almost surely as  $n \rightarrow \infty$ . We reach the conclusion of the theorem.

**PROOF OF THEOREM B.7** The proof argument of this theorem is a direct application of the results with uniform laws of large numbers in the proof of Theorem B.6. In fact, we define

$$\begin{aligned} \tilde{\mathcal{F}}_0(\epsilon) = & \left\{ (S, \{\pi_y\}, \{b(\ell)\}) : S = \left\{ h(y, z; 0) : (y, z) \in \mathcal{J} \right\}, \right. \\ & \text{and } \left. \inf_{(\tilde{S}_0, \{\tilde{\pi}_y^0\}, \{\tilde{b}_0(\ell)\}) \in \mathcal{G}(\tilde{\mathcal{F}}_0)} \left\{ H(S, \tilde{S}_0) + \sum_{y=1}^K |\pi_y - \tilde{\pi}_y^0| + \sum_{z \in \mathcal{L}} \|b(\ell) - \tilde{b}_0(\ell)\| \right\} \geq \epsilon \right\} \end{aligned}$$

for any  $\epsilon > 0$ . Since the parameter spaces of  $\theta$  are compact sets, the set  $\tilde{\mathcal{F}}_0(\epsilon)$  is also a compact set for all  $\epsilon > 0$ . Denote

$$\begin{aligned} & g(S, \{\pi_y\}, \{b(\ell)\}) \\ = & \alpha_{\text{RC}} \left\{ \bar{\lambda} \left( \int \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \left( h^\top(y, z; 0)x + \tau(\theta_{y, z}) \right) \right\} \left( \frac{\|x - h(y', z'; 0)\|^2}{2} - \log(\pi_{y', z'}) \right) dP(x) \right) \right. \\ & \left. + (1 - \bar{\lambda}) \left( \int \sum_{z' \in \mathcal{L}} 1 \left\{ z' = \arg \max_{z \in \mathcal{L}} \left( h^\top(y, z; 0)x + \tau(\theta_{y, z}) \right) \right\} \left( \frac{\|x - h(y, z'; 0)\|^2}{2} - \log(\pi_{y, z'}) \right) dQ(x, y) \right) \right\} \\ & - \alpha_{\text{CE}} \int \log q_\theta(y|x) dQ(x, y) + \alpha_{\text{KL}} \int \sum_{y=1}^K q_\theta(y|x) dQ(x, c) \log \left( \frac{q_\theta(y|x)}{\pi_y} \right) dP(x) \end{aligned}$$

for all  $(S, \{\pi_y\}, \{b(\ell)\})$ . From the definition of  $\tilde{\mathcal{F}}_0(\epsilon)$ , we have that

$$g(S, \{\pi_y\}, \{b(\ell)\}) > g(\tilde{S}_0, \{\tilde{\pi}_y^0\}, \{\tilde{b}_0(\ell)\})$$

for all  $(S, \{\pi_y\}, \{b(\ell)\}) \in \tilde{\mathcal{F}}_0(\epsilon)$  and  $(\tilde{S}_0, \{\tilde{\pi}_y^0\}, \{\tilde{b}_0(\ell)\}) \in \mathcal{G}(\tilde{\mathcal{F}}_0)$ . As  $\tilde{\mathcal{F}}_0(\epsilon)$  is a compact set, we further have that

$$\inf_{(S, \{\pi_y\}, \{b(\ell)\}) \in \tilde{\mathcal{F}}_0(\epsilon)} g(S, \{\pi_y\}, \{b(\ell)\}) > g(\tilde{S}_0, \{\tilde{\pi}_y^0\}, \{\tilde{b}_0(\ell)\})$$

for all  $(\tilde{S}_0, \{\tilde{\pi}_y^0\}, \{\tilde{b}_0(\ell)\}) \in \mathcal{G}(\tilde{\mathcal{F}}_0)$  and  $\epsilon > 0$ .

Now, according to the uniform laws of large numbers established in the proof of Theorem B.6, we have that  $Y_n \rightarrow g(\tilde{S}_n, \{\tilde{\pi}_y\}, \{\tilde{b}(\ell)\})$  almost surely as  $n \rightarrow \infty$ . According to the result of Theorem B.6, it implies that  $g(\tilde{S}_n, \{\tilde{\pi}_y\}, \{\tilde{b}(\ell)\}) \rightarrow g(\tilde{S}_0, \{\tilde{\pi}_y^0\}, \{\tilde{b}_0(\ell)\})$  almost surely for all  $(\tilde{S}_0, \{\tilde{\pi}_y^0\}, \{\tilde{b}_0(\ell)\}) \in \mathcal{G}(\tilde{\mathcal{F}}_0)$ . Therefore, for each  $\epsilon > 0$  we can find sufficiently large  $N$  such that we have

$$\inf_{(\tilde{S}_0, \{\tilde{\pi}_y^0\}, \{\tilde{b}_0(\ell)\}) \in \mathcal{G}(\tilde{\mathcal{F}}_0)} \left\{ H(\tilde{S}_n, \tilde{S}_0) + \sum_{y=1}^K |\tilde{\pi}_y - \tilde{\pi}_y^0| + \sum_{z \in \mathcal{L}} \|\tilde{b}(\ell) - \tilde{b}_0(\ell)\| \right\} < \epsilon$$

almost surely for all  $n \geq N$ . As a consequence, we achieve the conclusion of the theorem.

**PROOF OF THEOREM B.8** The proof of the theorem is an application of Theorem 11 for generalization bound with margin from [60] based on an evaluation of Rademacher complexity. In particular, we denote

$$\begin{aligned}\mathcal{J}_n &= \left\{ h_{\bar{\tau}_n}(x, y) : \mathbb{R}^{D^{(0)}} \times \{1, \dots, K\} \rightarrow \mathbb{R} \right| \\ h_{\bar{\tau}_n}(x, y) &= \max_{z \in \mathcal{L}(\bar{\tau}_n)} \left\{ h^\top(y, z; 0)x + \tau(\theta_{y,z}) \right\} + \log \pi_y \quad \forall (x, y) \text{ for some } |\mathcal{L}(\bar{\tau}_n)| \leq \bar{\tau}_n |\mathcal{L}| \right\}.\end{aligned}$$

Now, we denote  $\tilde{\mathcal{J}}_n = \{h_{\bar{\tau}_n}(\cdot, y) : y \in \{1, \dots, K\}, h_{\bar{\tau}_n} \in \mathcal{J}_n\}$ . For any  $\delta > 0$ , using the same argument as that of the proof of Theorem 11 in [60], with probability at least  $1 - \delta$ , we have

$$\begin{aligned}R_0(f_{\bar{\tau}_n}) &\leq \inf_{\Gamma \in (0,1]} \left\{ R_{n,\Gamma}(f_{\bar{\tau}_n}) + \frac{8K(2K-1)}{\Gamma} \mathfrak{R}_n(\tilde{\mathcal{J}}_n) \right. \\ &\quad \left. + \left( \frac{\log \log_2(2\Gamma^{-1})}{n} \right)^{1/2} + \sqrt{\frac{\log(2\delta^{-1})}{2n}} \right\}\end{aligned}\quad (45)$$

where  $\mathfrak{R}_n(\tilde{\mathcal{J}}_n)$  is Rademacher complexity of  $\tilde{\mathcal{J}}_n$ , which in our case is defined as

$$\mathfrak{R}_n(\tilde{\mathcal{J}}_n) = \mathbb{E} \sup_{\theta} \sup_{|\mathcal{L}(\bar{\tau}_n)| \leq \bar{\tau}_n |\mathcal{L}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \left( \max_{g \in \mathcal{L}(\bar{\tau}_n)} \left\{ h^\top(y, z; 0)x_i + \tau(\theta_{y,z}) \right\} + \log \pi_y \right) \right|$$

where  $\sigma_1, \dots, \sigma_n$  are i.i.d. Rademacher random variables. Since  $\bar{\gamma}$  is the lower bound of  $\pi_y$  for all  $1 \leq y \leq K$ , we obtain that

$$\begin{aligned}\mathfrak{R}_n(\tilde{\mathcal{J}}) &\leq \mathbb{E} \sup_{\theta} \sup_{|\mathcal{L}(\bar{\tau}_n)| \leq \bar{\tau}_n |\mathcal{L}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \left( \max_{g \in \mathcal{L}(\bar{\tau}_n)} \left\{ h^\top(y, z; 0)x_i + \tau(\theta_{y,z}) \right\} \right) \right| \\ &\quad + \mathbb{E} \sup_{\theta} \sup_{|\mathcal{L}(\bar{\tau}_n)| \leq \bar{\tau}_n |\mathcal{L}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \log \pi_y \right| \\ &\leq \mathbb{E} \sup_{\theta} \sup_{|\mathcal{L}(\bar{\tau}_n)| \leq \bar{\tau}_n |\mathcal{L}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \left( \max_{g \in \mathcal{L}(\bar{\tau}_n)} \left\{ h^\top(y, z; 0)x_i + \tau(\theta_{y,z}) \right\} \right) \right| + \frac{|\log \bar{\gamma}|}{\sqrt{n}}\end{aligned}$$

Furthermore, we have the following inequalities

$$\begin{aligned}\mathbb{E} \sup_{\theta} \sup_{|\mathcal{L}(\bar{\tau}_n)| \leq \bar{\tau}_n |\mathcal{L}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \left( \max_{g \in \mathcal{L}(\bar{\tau}_n)} \left\{ h^\top(y, z; 0)x_i + \tau(\theta_{y,z}) \right\} \right) \right| \\ \leq \mathbb{E} \sup_{|S'| \leq \bar{c}_n |\mathcal{L}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \max_{s \in S'} s^\top [x_i, 1] \right| \\ \leq 2\bar{c}_n |\mathcal{L}| \mathbb{E} \sup_{s \in \mathbb{B}(R)} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i s^\top [x_i, 1] \right| \\ \leq 2\bar{c}_n |\mathcal{L}| R \mathbb{E} \left\| \frac{1}{n} \sum_{i=1}^n \sigma_i [x_i, 1] \right\| \\ \leq \frac{2\bar{c}_n |\mathcal{L}| (R^2 + 1)}{\sqrt{n}}\end{aligned}$$

where the final inequality is due to Cauchy-Schwartz's inequality. Combining the above results with (45), we achieve the conclusion of the theorem.

## D Appendix D

In this appendix, we provide further extensions of the main results in Appendix A and Appendix B.

### D.1 Additional discussion with NRM and cross entropy

Thus far, we have established the lower bound and the upper bound of maximizing the conditional log likelihood in terms of the cross entropy in Theorem B.1. In the following full theorem, we will demonstrate that this cross entropy also the upper bound of maximizing the full posterior of NRM.

**Theorem D.1.** *Denote  $\mathcal{A}_\gamma = \{\theta : \|h(y, z; 0)\| = \gamma\}$  for any  $\gamma > 0$ . For any  $n \geq 1$  and  $\sigma > 0$ , let  $x_1, \dots, x_n$  be i.i.d. samples from the NRM. Then, the following holds*

(a) *Upper bound:*

$$\begin{aligned} \max_{(z_i)_{i=1}^n, \theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log p(y_i, z_i | x_i; \theta) \\ \leq \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log \left( \text{Softmax} \left( \max_{z_i} \left( \frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2} \right) + b_{y_i} \right) \right) \\ = \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log q(y_i | x_i) = - \min_{\theta \in \mathcal{A}_\gamma} H_{p,q}(y|x) \end{aligned}$$

where  $b_y = \log \pi_y$  for all  $1 \leq y \leq K$ ,  $q(y|x) = \text{Softmax} \left( \max_z \left( h^\top(y, z; 0)x + \eta(y, z) \right) / \sigma^2 + b_y \right)$  for all  $(x, y)$ , and  $H_{p,q}(y|x)$  is the cross-entropy between the estimated posterior  $q(y|x)$  and the true posterior given by the ground-true labels  $p(y|x)$ .

(b) *Lower bound:*

$$\max_{(z_i)_{i=1}^n, \theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log p(y_i, z_i | x_i; \theta) \geq -\log(|\mathcal{L}|) - \min_{\theta \in \mathcal{A}_\gamma} H_{p,q}(y|x)$$

where  $|\mathcal{L}|$  denotes the total number of possible rendering paths with a given label.

*Remark D.2.* Combining with the bounds in Theorem B.1, we obtain that

$$\max_{(z_i)_{i=1}^n, \theta \in \mathcal{A}_\gamma} \sum_{i=1}^n \ln p(y_i | x_i, z_i; \theta) \geq - \min_{\theta} H_{p,q}(y|x) \geq \max_{(z_i)_{i=1}^n, \theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log p(y_i, z_i | x_i; \theta).$$

Therefore, the cross entropy loss is the lower bound and the upper bound of the conditional log likelihood and the full posterior of NRM respectively.

### D.2 Relaxation of constant norm with rendered images

So far, we have assumed constant norm of rendered image  $h(y, z; 0)$  to derive the results in part (a) of Theorem 3.3 and Theorem D.1. We would like to emphasize that this assumption is just for the simplicity of the argument and the elegance of the statements in these results. Without this assumption, the results of these theorems may need to change as follows. We denote  $\mathcal{B}_{M_1, M_2} := \{\theta : M_1 \leq \|h(y, z; 0)\| \leq M_2\}$  for some given non-negative constants  $M_1 < M_2$ . Here,  $\theta \in \mathcal{B}_{M_1, M_2}$  does not need to satisfy the non-negativity assumption. Then, we obtain that



**Theorem D.3.** (*Relaxation of Theorem B.1*) For any  $n \geq 1$  and  $\sigma > 0$ , let  $x_1, \dots, x_n$  be i.i.d. samples from the NRM. Given the formulation of  $\mathcal{B}_{M_1, M_2}$ , the following holds  
(a) Lower bound:

$$\begin{aligned}
& \max_{(z_i)_{i=1}^n, \theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \ln p(y_i | x_i, z_i; \theta) \\
& \geq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log \left( \text{Softmax} \left( g(y_i, x_i) \right) + b_{y_i} \right) - \frac{M_2^2 - M_1^2}{2\sigma^2} \\
& = \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log \tilde{q}(y_i | x_i) - \frac{M_2^2 - M_1^2}{2\sigma^2} \\
& = - \min_{\theta \in \mathcal{B}_{M_1, M_2}} H_{p, \tilde{q}}(y | x) - \frac{M_2^2 - M_1^2}{2\sigma^2}
\end{aligned}$$

where  $b_y = \log \pi_y$  as  $1 \leq y \leq K$ ,  $\tilde{q}(y | x) = \text{Softmax} \left( h(y, x) + b_y \right)$  for all  $(x, y)$ , and

$$g(y, x) := \frac{1}{\sigma^2} h^\top(y) \text{MaxPool} \left( \text{ReLU} \left( \text{Conv} \left( \Gamma(\ell), \dots \text{MaxPool} \left( \text{ReLU} \left( \text{Conv} \left( \Gamma(1), x \right) + b(1) \right) \right) \right) \right) \right) \dots + b(\ell) \Big)$$

for all  $(x, y)$ .

(b) Upper bound:

$$\begin{aligned}
& \max_{(z_i)_{i=1}^n, \theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \ln p(y_i | x_i, z_i; \theta) \\
& \leq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \left\{ \log \left( \text{Softmax} \left( g(y_i, x_i) \right) + b_{y_i} \right) + \underbrace{\left\{ \max_{z_i} \left( \frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2} \right) - g(y_i, x_i) \right\}}_{\text{CNN loss}} \right\} \\
& \quad + \max_y \left( g(y, x_i) - \frac{h^\top(y, \bar{z}_i; 0)x_i + \eta(y, \bar{z}_i)}{\sigma^2} \right) + \log K + \frac{M_2^2 - M_1^2}{2\sigma^2}
\end{aligned}$$

where  $\bar{z}_i = \arg \max_{z_i} p(y_i | x_i, z_i; \theta)$  for  $1 \leq i \leq n$ .

*Remark D.4.* Comparing to the lower bound in Theorem B.1, the cross entropy term  $H_{p, \tilde{q}}$  has a direct connection to CNN without relying on the non-negativity assumption. Additionally, we need to pay a price of  $(M_2^2 - M_1^2)/(2\sigma^2)$  as we relax the constant norm assumption with rendering paths  $h(y, z; 0)$ . Last but not least, comparing to the upper bound in Theorem B.1, apart from the inclusion of  $(M_2^2 - M_1^2)/(2\sigma^2)$ , we also need to pay an additional price of CNN loss as we directly connect the CNN structure to the cross entropy term  $H_{p, \tilde{q}}$  without hinging on the non-negativity assumption.

Similar the the changes with results of Theorem D.3, we also have the following modifications regarding the upper bound and lower bound of full posterior of NRM under the relaxation of constant norm with rendered images  $h(y, z; 0)$ .

**Theorem D.5.** (Relaxation of Theorem D.1) For any  $n \geq 1$  and  $\sigma > 0$ , let  $x_1, \dots, x_n$  be i.i.d. samples from the NRM. Given the formulation of  $\mathcal{B}_{M_1, M_2}$ , the following holds

(a) Upper bound:

$$\begin{aligned}
& \max_{(z_i)_{i=1}^n, \theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log p(y_i, z_i | x_i; \theta) \\
& \leq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \left\{ \log \left( \text{Softmax} \left( g(y_i, x_i) \right) + b_{y_i} \right) + \underbrace{\left\{ \max_{z_i} \left( \frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2} \right) - g(y_i, x_i) \right\}}_{\text{CNN loss}} \right\} \\
& \quad + \frac{M_2^2 - M_1^2}{2\sigma^2} \\
& = \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log \tilde{q}(y_i | x_i) + \left\{ \max_{z_i} \left( \frac{h^\top(y_i, z_i; 0)x_i + \eta(y, z_i)}{\sigma^2} \right) - g(y_i, x_i) \right\} + \frac{M_2^2 - M_1^2}{2\sigma^2} \\
& = - \min_{\theta \in \mathcal{B}_{M_1, M_2}} H_{p, \tilde{q}}(y | x) + \left\{ \max_{z_i} \left( \frac{h^\top(y_i, z_i; 0)x_i + \eta(y, z_i)}{\sigma^2} \right) - g(y_i, x_i) \right\} + \frac{M_2^2 - M_1^2}{2\sigma^2}
\end{aligned}$$

where  $b_y, \tilde{q}(y|x)$ , and  $g(y, x)$  are defined as in Theorem D.3.

(b) Lower bound:

$$\begin{aligned}
& \max_{(z_i)_{i=1}^n, \theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log p(y_i, z_i | x_i; \theta) \\
& \geq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \left\{ \log \left( \text{Softmax} \left( g(y_i, x_i) \right) + b_{y_i} \right) \right. \\
& \quad \left. + \left( \max_y g(y, x_i) - \max_{y, z} \left( \frac{h^\top(y, z; 0)x_i + \eta(y, z)}{\sigma^2} \right) \right) \right\} - \log(|\mathcal{J}|) - \frac{M_2^2 - M_1^2}{2\sigma^2}
\end{aligned}$$

*Remark D.6.* Comparing to the upper bound in Theorem D.1, we have the additional CNN loss and the term  $(M_2^2 - M_1^2)/(2\sigma^2)$  as we relax the constant norm with rendered image  $h(y, z; 0)$  as well as the non-negativity assumption to CNN structure with the cross entropy term. Moreover, in addition to the price of CNN loss and of the term  $(M_2^2 - M_1^2)/(2\sigma^2)$ , we also need to pay the price of  $\log(|\mathcal{J}|)$  instead of  $\log(|\mathcal{L}|)$  in Theorem D.1.

### D.3 Statistical guarantees for unsupervised NRM without noise

As being mentioned in Appendix B, under the non-negativity assumption of the intermediate rendered images  $h(y, z; \ell)$ , the tractable relaxation of objective function of unsupervised NRM without noise can be formulated as

$$V_n := \min_{\theta} \frac{1}{n} \sum_{i=1}^n \sum_{(y', z') \in \mathcal{J}} \mathbb{1} \left\{ (y', z') = \arg \max_{(y, z; 0) \in \mathcal{J}} \left( h^\top(y, z; 0)x_i + \tau(\theta_{y, z}) \right) \right\} \left( \frac{\|x_i - h(y', z'; 0)\|^2}{2} - \log(p_{y', z'}) \right) \quad (46)$$

where  $h(y, z; 0) = \Lambda(z; 1) \dots \Lambda(z; L)\mu(y)$ ,  $\Lambda(z; l) = \sum_{p \in \mathcal{P}(\ell)} s(\ell, p)T(t; \ell, p)B(\ell, p)\Gamma(\ell, p)$ , and  $p_{y', z'} = \exp \left( \tau(\theta_{y', z'}) + \log \pi_{y'} \right) / \left( \sum_{y=1}^K \exp \left( \max_{z \in \mathcal{L}} \tau(\theta_{y, z}) + \log \pi_y \right) \right)$ . For each  $\theta$ , we denote  $S = S(\theta) =$

$\{h(y, z; 0) : (y, z) \in \mathcal{J}\}$  the set of all possible *rendered images* associated with  $\theta$ . Since the rendered images in  $S$  can be identical, it is clear that the number of elements of  $S$ , which is defined as  $|S|$ , is upper bounded by a fixed  $|\mathcal{J}|$  total number of rendered images, which depends on the number of layers  $L$  and the number of pixels in each layer. Similar to the semi-supervised setting in (16), a notable property of  $S$  is that all of their rendered images share the parameters  $\Gamma(\ell)$  and  $\mu(y)$  for all  $1 \leq \ell \leq L$  and  $1 \leq y \leq K$ . Therefore, we call the above objective function *shared parameters latent dependence regularized K-means (SPLD regularized K-means)*.

**Existence of optimal solutions** When there are no constraints with the parameter spaces  $\Theta_\ell$ ,  $\Omega$ , and  $\Xi_l$  of  $\mu(y)$ ,  $\Gamma(\ell)$ , and  $b(\ell)$  as  $1 \leq y \leq K$ ,  $1 \leq \ell \leq L$ , and  $g \in \mathcal{L}$ , the existence of optimal solutions of objective function (46) is not guaranteed. To ensure this existence, we will impose the compact constraints on these parameter spaces.

**Lemma D.7.** *Assume that  $\Theta_\ell$  is a compact subset of  $\mathbb{R}^{F(\ell) \times D(\ell)}$ ,  $\Omega$  is a compact subset of  $\mathbb{R}^{D(\ell)}$ , and  $\Xi_l$  is a compact subset of  $\mathbb{R}^{D(\ell)}$  for  $1 \leq \ell \leq L$  under Frobenius norm. Then, there exist optimal solutions  $\hat{\theta} = \left( \{\hat{\mu}(y)\}_{y=1}^K, \{\hat{\Gamma}(\ell)\}_{\ell=1}^L, \{\hat{\pi}_y\}_{y=1}^K, \{\hat{b}(\ell)\}_{\ell=1}^L \right)$  of SPLD regularized K-means.*

To simplify the presentation later, we will assume throughout this section the following key assumptions:

- The parameter spaces  $\Theta_\ell$ ,  $\Omega$ ,  $\Xi_l$  are compact subsets of  $\mathbb{R}^{F(\ell) \times D(\ell)}$ ,  $\mathbb{R}^{D(\ell)}$ , and  $\mathbb{R}^{D(\ell)}$  respectively for  $1 \leq \ell \leq L$ .
- The prior probability  $\pi = \{\pi_y\}_{y=1}^K$  of labels in  $\theta$  is such that  $\pi_y > \bar{\gamma}$  as  $1 \leq y \leq K$  for some given sufficiently small  $\bar{\gamma} > 0$ .

The assumption with prior probability  $\pi$  is to guarantee that the value of  $\log(p_{y,z})$  will not go to  $-\infty$  as the value of  $\pi_y$  becomes close to 0.

**Consistency of SPLD regularized K-means** Now, we would like to establish the consistency of the objective function and optimal solutions of SPLD regularized K-means. We assume that  $x_1, \dots, x_n$  are i.i.d. samples from a true but unknown distribution  $P$  with finite second moment, i.e.,  $\int \|x\|^2 dP(x) < \infty$ .

**Theorem D.8.** *(Consistency of objective function of SPLD regularized K-means) Assume that  $\mathbb{P}(\|I\| \leq R) = 1$  as  $I \sim P$  for some given  $R > 0$ . We denote the population version of SPLD regularized K-means as follows*

$$\bar{V} := \min_{\theta} \int \sum_{(y', z') \in \mathcal{J}} 1_{\left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0)x + \tau(\theta_{y, z}) \right) \right\}} \left( \frac{\|x - h(y', z'; 0)\|^2}{2} - \log(p_{y', z'}) \right) dP(x).$$

Then, we obtain that  $V_n \rightarrow \bar{V}$  almost surely as  $n \rightarrow \infty$ .

So far, we have established the convergence of objective function of SPLD K-means to the objective function of its corresponding population version. Now, we would like to study the convergence of optimal rendered images and optimal solutions of SPLD regularized K-means to those of population SPLD regularized K-means. In particular, we denote  $\mathcal{F}_0$  the set of all optimal solutions

$\theta^0 = \left( \{\mu^0(y)\}_{y=1}^K, \{\Gamma_0(\ell)\}_{\ell=1}^L, \{\pi_y^0\}_{y=1}^K, \{b_0(\ell)\}_{\ell=1}^L \right)$  of population SPLD regularized K-means. For each  $\theta^0 \in \mathcal{F}_0$ , we define  $S_0$  the set of optimal rendered images associated with  $\theta^0$ . We denote  $\mathcal{G}(\mathcal{F}_0)$  the corresponding set of all optimal rendered images  $S_0$ , optimal prior probabilities  $\{\pi_y^0\}_{y=1}^K$ , and optimal biases  $\{b_0(\ell)\}_{\ell=1}^L$ .

**Theorem D.9.** (*Consistency of optimal rendering paths and optimal solutions of SPLD regularized K-means*) Assume that  $\mathbb{P}(\|x\| \leq R) = 1$  as  $x \sim P$  for some given  $R > 0$ . Then, we obtain that

$$\inf_{(S_0, \{\pi_y^0\}, \{b_0(\ell)\}) \in \mathcal{G}(\mathcal{F}_0)} \left\{ H(\hat{S}_n, S_0) + \sum_{y=1}^K |\hat{\pi}_y - \pi_y^0| + \sum_{\ell=1}^L \|\hat{b}(\ell) - b_0(\ell)\| \right\} \rightarrow 0$$

almost surely as  $n \rightarrow \infty$ . Here,  $H(\cdot, \cdot)$  stands for the Hausdorff metric.

**Convergence rate with clustering risk of SPLD regularized K-means** Remind that we denote  $\hat{\theta} := \left( \{\hat{\mu}(y)\}_{y=1}^K, \{\hat{\Gamma}(\ell)\}_{\ell=1}^L, \{\hat{\pi}_y\}_{y=1}^K, \{\hat{b}(\ell)\}_{\ell=1}^L \right)$  the optimal solutions of SPLD regularized K-means. These optimal solutions lead to the corresponding set of optimal rendered images  $\hat{S}_n = \{\hat{h}(y, z; 0) : (y, z) \in \mathcal{J}\}$  where

$$\begin{aligned} \hat{h}(y, z; 0) &:= \hat{\Lambda}(z; 1) \dots \hat{\Lambda}(z; L) \hat{\mu}(y), \\ \hat{\Lambda}(z; \ell) &:= \sum_{p \in \mathcal{P}(\ell)} s(\ell, p) T(t; \ell, p) B(\ell, p) \hat{\Gamma}(\ell, p) \end{aligned}$$

for all  $1 \leq \ell \leq L$ . In practice, only a small fraction of optimal rendered images in  $\hat{S}_n$  is indeed useful for clustering the data  $x_1, \dots, x_n$ . These optimal rendered images correspond to the optimal *active rendering paths* of  $\mathcal{J}$ . In particular, we denote a smallest subset  $\bar{\mathcal{J}}_n$  of  $\mathcal{J}$  such that  $|\bar{\mathcal{J}}_n| \leq \bar{c}_n |\mathcal{J}|$  and the following holds

$$V_n = \frac{1}{n} \sum_{i=1}^n \sum_{(y', z') \in \bar{\mathcal{J}}_n} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \bar{\mathcal{J}}_n} \left( \hat{h}(y, z; 0)^\top x_i + \tau(\hat{\theta}_{y, z}) \right) \right\} \left( \frac{\|x_i - \hat{h}(y', z'; 0)\|^2}{2} - \log(\hat{p}(y', z')) \right).$$

where  $\hat{p}(y', z') = \exp\left(\tau(\hat{\theta}_{y', z'}) + \log \hat{\pi}_{y'}\right) / \left(\sum_{y=1}^K \exp\left(\max_{z \in \mathcal{L}} \tau(\hat{\theta}_{y, z}) + \log \hat{\pi}_y\right)\right)$ . The set  $\bar{\mathcal{J}}_n$  is called the set of optimal active paths of  $\mathcal{J}$  while the positive number  $\bar{c}_n \leq 1$  is called the ratio of active rendering paths. Throughout this section, we assume that  $\bar{c}_n$  is independent of  $x_1, \dots, x_n$  and only depends on sample size  $n$ .

Inspired by the idea of generalization gap in classification setting, the measure that we utilize to evaluate the convergence rate in clustering is the difference between the objective function of SPLD regularized K-means and its corresponding clustering risk, which can be defined as follows

$$\bar{V}_n = \int \left( \sum_{(y', z') \in \bar{\mathcal{J}}_n} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \bar{\mathcal{J}}_n} \left( \hat{h}(y, z; 0)^\top x + \tau(\hat{\theta}_{y, z}) \right) \right\} \left( \frac{\|x - \hat{h}(y, z; 0)\|^2}{2} - \log(\hat{p}(y', z')) \right) \right) dP(x).$$

for all  $n \geq 1$ . We have the following result establishing the generalization gap between  $V_n$  and  $\bar{V}_n$  based on the number of active rendering paths in SPLD regularized K-means.

**Theorem D.10.** *(Convergence rate of SPLD regularized K-means) Assume that  $\mathbb{P}(\|x\| \leq R) = 1$  as  $x \sim P$  for some given  $R > 0$ . Additionally, the parameter spaces  $\Theta_\ell$  and  $\Omega$  are chosen such that  $\|h(y, z; 0)\| \leq R$  for all  $(y, z) \in \mathcal{J}$ . Then, the following holds*

$$\left| \mathbb{E}V_n - \mathbb{E}\bar{V}_n \right| \leq \frac{4\bar{c}_n|\mathcal{J}|R^2}{\sqrt{n}} + (2\log(1/\bar{\gamma}) + R^2) \sqrt{\frac{2(\log 8 + (D^{(0)} + 1)(\bar{c}_n|\mathcal{J}| - 1)\log n)}{n}}$$

where  $\bar{\gamma}$  is a given lower bound for the prior probability  $\pi_y$  as  $1 \leq y \leq K$ .

The proof of Theorem D.10 is deferred to Appendix E.

## E Appendix E

In this appendix, we provide detail proofs for the remaining results in Appendix D.

**PROOF OF THEOREM D.1** (a) Regarding the lower bound, we obtain that

$$\begin{aligned}
& \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log p(y_i, z_i | x_i; \theta) \\
&= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log \frac{p(x_i | y_i, z_i; \theta) p(y_i, z_i; \theta)}{\sum_{y, z} p(x_i | y, z; \theta) p(y, z; \theta)} \\
&\stackrel{(a)}{\leq} \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log \frac{p(x_i | y_i, z_i; \theta) p(y_i, z_i; \theta)}{\sum_{y=1}^K \max_z p(x_i | y, z; \theta) p(y, z; \theta)} \\
&= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log \left( \text{Softmax} \left( \max_{z_i} \psi_i(y_i, z_i) \right) \right) \\
&= \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log q(y_i | x_i) = - \min_{\theta \in \mathcal{A}_\gamma} H_{p, q}(y | x)
\end{aligned}$$

where the inequality in (a) is due to the fact that

$$\sum_{y, z} p(x_i | y, z; \theta) p(y, z; \theta) \geq \sum_{y=1}^K \max_z p(x_i | y, z; \theta) p(y, z; \theta)$$

for all  $1 \leq i \leq n$ . Therefore, we achieve the conclusion of part (a) of the theorem.

(b) Regarding the upper bound, according to the above formulations, we also have

$$\begin{aligned}
& \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log p(y_i, z_i | x_i; \theta) \\
&\geq \max_{\theta \in \mathcal{A}_\gamma} \frac{1}{n} \sum_{i=1}^n \log \frac{\max_{z_i} p(x_i | y_i, z_i; \theta) p(y_i, z_i; \theta)}{|\mathcal{L}| \sum_y \max_z p(x_i | y, z; \theta) p(y, z; \theta)} \\
&= -\log(|\mathcal{L}|) - \min_{\theta \in \mathcal{A}_\gamma} H_{p, q}(y | x)
\end{aligned}$$

where  $|\mathcal{L}|$  denote the total number of possible paths with a given label. Note that, the above inequality is due to the simple upper bound

$$\sum_{y, z} p(x_i | y, z; \theta) p(y, z; \theta) \leq |\mathcal{L}| \sum_y \max_z p(x_i | y, z; \theta) p(y, z; \theta)$$

for all  $1 \leq i \leq n$ .

**PROOF OF THEOREM D.3** (a) Using the same argument as that in the proof of Theorem B.1, we obtain the following equation

$$\begin{aligned} & \max_{(z_i)_{i=1}^n, \theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log p(y_i | x_i, z_i; \theta) \\ &= \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log \left( \frac{p(x_i | y_i, z_i; \theta) \exp\left(\frac{\eta(y_i, z_i)}{\sigma^2}\right) \pi_{y_i}}{\sum_{y=1}^K p(x_i | y, z_i; \theta) \exp\left(\frac{\eta(y, z_i)}{\sigma^2}\right) \pi_y} \right) \end{aligned}$$

From the assumption that  $M_1 \leq \|h(y, z_i; 0)\| \leq M_2$ , for any  $(y, z_i)$ , it is clear that

$$p(x_i | y, z_i; \theta) \exp\left(\frac{\eta(y, z_i)}{\sigma^2}\right) \pi_y \leq \frac{1}{(\sqrt{2\pi}\sigma)^{D(0)}} \exp\left(-\frac{\|x_i\|^2 - 2h^\top(y, z_i; 0)x_i + M_1^2 - 2\eta(y, z_i)}{2\sigma^2}\right) \pi_y$$

and

$$p(x_i | y, z_i; \theta) \exp\left(\frac{\eta(y, z_i)}{\sigma^2}\right) \pi_y \geq \frac{1}{(\sqrt{2\pi}\sigma)^{D(0)}} \exp\left(-\frac{\|x_i\|^2 - 2h^\top(y, z_i; 0)x_i + M_2^2 - 2\eta(y, z_i)}{2\sigma^2}\right) \pi_y.$$

Therefore, we achieve that

$$\begin{aligned} & \max_{(z_i)_{i=1}^n, \theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log p(y_i | x_i, z_i; \theta) \\ & \geq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log \left( \frac{\exp\left(\frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2}\right) \pi_{y_i}}{\sum_{y=1}^K \exp\left(\frac{h^\top(y, z_i; 0)x_i + \eta(y, z_i)}{\sigma^2}\right) \pi_y} \right) - \frac{M_2^2 - M_1^2}{2\sigma^2} \\ & \geq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log \left( \text{Softmax}\left(g(y_i, x_i)\right) + b_{y_i} \right) - \frac{M_2^2 - M_1^2}{2\sigma^2} \\ & = \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log \tilde{q}(y_i | x_i) - \frac{M_2^2 - M_1^2}{2\sigma^2} = - \min_{\theta \in \mathcal{B}_{M_1, M_2}} H_{p, \tilde{q}}(y | x) - \frac{M_2^2 - M_1^2}{2\sigma^2} \end{aligned}$$

As a consequence, we achieve the conclusion of part (a) of the theorem.

(b) Regarding the upper bound, we have

$$\begin{aligned} & \max_{(z_i)_{i=1}^n, \theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log p(y_i | x_i, z_i; \theta) \\ & \leq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log \left( \frac{\exp\left(\frac{h^\top(y_i, \bar{z}_i; 0)x_i + \eta(y_i, \bar{z}_i)}{\sigma^2}\right) \pi_{y_i}}{\sum_{y=1}^K \exp\left(\frac{h^\top(y, \bar{z}_i; 0)x_i + \eta(y, \bar{z}_i)}{\sigma^2}\right) \pi_y} \right) + \frac{M_2^2 - M_1^2}{2\sigma^2} \\ & \leq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log \left( \frac{\exp\left(\max_{z_i} \frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2}\right) \pi_{y_i}}{\sum_{y=1}^K \exp\left(\frac{h^\top(y, \bar{z}_i; 0)x_i + \eta(y, \bar{z}_i)}{\sigma^2}\right) \pi_y} \right) + \frac{M_2^2 - M_1^2}{2\sigma^2} \quad (47) \end{aligned}$$

Additionally, denote  $\tilde{z}_i$  the maximum value of  $z$  in the CNN structure of  $h(y, x_i)$  for all  $1 \leq y \leq K$  and  $1 \leq i \leq n$ . Then, the following inequality holds

$$\begin{aligned}
& \log \left( \frac{\exp \left( \max_{z_i} \frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2} \right) \pi_{y_i}}{\sum_{y=1}^K \exp \left( \frac{h^\top(y, \bar{z}_i; 0)x_i + \eta(y, \bar{z}_i)}{\sigma^2} \right) \pi_y} \right) - \log \left( \text{Softmax} \left( g(y_i, x_i) \right) + b_{y_i} \right) \\
&= \max_{z_i} \left( \frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2} \right) - g(y_i, x_i) + \log \left( \frac{\sum_{y=1}^K \exp \left( \frac{h^\top(y, \tilde{z}_i; 0)x_i + \eta(y, \tilde{z}_i)}{\sigma^2} \right) \pi_y}{\sum_{y=1}^K \exp \left( \frac{h^\top(y, \bar{z}_i; 0)x_i + \eta(y, \bar{z}_i)}{\sigma^2} \right) \pi_y} \right) \\
&\leq \max_{z_i} \left( \frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2} \right) - g(y_i, x_i) + \max_y \left\{ \left( \frac{h^\top(y, \tilde{z}_i; 0)x_i + \eta(y, \tilde{z}_i)}{\sigma^2} \right) + \log \pi_y \right\} \\
&\quad - \max_y \left\{ \left( \frac{h^\top(y, \bar{z}_i; 0)x_i + \eta(y, \bar{z}_i)}{\sigma^2} \right) + \log \pi_y \right\} + \log K \\
&\leq \max_{z_i} \left( \frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2} \right) - g(y_i, x_i) + \max_y \left\{ \left( g(y, x_i) - \frac{h^\top(y, \bar{z}_i; 0)x_i + \eta(y, \bar{z}_i)}{\sigma^2} \right) \right\} + \log K
\end{aligned} \tag{48}$$

Combining the results from (47) and (48), we obtain the conclusion of part (b) of the theorem.

**PROOF OF THEOREM D.5** To simplify the presentation, throughout this proof, we denote

$A = \max_{(z_i)_{i=1}^n, \theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log p(y_i, z_i | x_i; \theta)$ . (a) Regarding the lower bound, we have

$$\begin{aligned}
A &= \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log \frac{p(x_i | y_i, z_i; \theta) p(y_i, z_i; \theta)}{\sum_{y, z} p(x_i | y, z; \theta) p(y, z; \theta)} \\
&\leq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \max_{z_i} \log \left( \frac{\exp \left( \frac{h^\top(y_i, z_i; 0)x_i + \eta(y_i, z_i)}{\sigma^2} \right) \pi_{y_i}}{\sum_{y=1}^K \exp(g(y, x_i)) \pi_y} \right) + \frac{M_2^2 - M_1^2}{2\sigma^2} \\
&= \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \left\{ \log \left( \text{Softmax} \left( g(y_i, x_i) \right) + b_{y_i} \right) \right. \\
&\quad \left. + \max_{z_i} \left( \frac{h^\top(y_i, z_i)x_i + \eta(y_i, z_i)}{\sigma^2} \right) - g(y_i, x_i) \right\} + \frac{M_2^2 - M_1^2}{2\sigma^2}
\end{aligned}$$

Therefore, we obtain the conclusion of part (a) of the theorem.



(b) Regarding the upper bound, we obtain that

$$\begin{aligned}
A &\geq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \log \left( \frac{\exp \left( \max_{z_i} \frac{h^\top(y_i, z_i)x_i + \eta(y_i, z_i)}{\sigma^2} \right) \pi_{y_i}}{\sum_{y, z} \exp \left( \frac{h^\top(y, z; 0)x_i + \eta(y, z)}{\sigma^2} \right) \pi_y} \right) - \frac{M_2^2 - M_1^2}{2\sigma^2} \\
&\geq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \left\{ \log \left( \text{Softmax} \left( g(y_i, x_i) \right) + b_{y_i} \right) \right. \\
&\quad \left. + \log \left( \frac{\sum_{y=1}^K \exp(g(y, x_i)) \pi_y}{\sum_{y, z} \exp \left( \frac{h^\top(y, z; 0)x_i + \eta(y, z)}{\sigma^2} \right) \pi_y} \right) \right\} - \frac{M_2^2 - M_1^2}{2\sigma^2} = B
\end{aligned} \tag{49}$$

By means of direction computation, the following holds

$$\begin{aligned}
B &\geq \max_{\theta \in \mathcal{B}_{M_1, M_2}} \frac{1}{n} \sum_{i=1}^n \left\{ \log \left( \text{Softmax} \left( g(y_i, x_i) \right) + b_{y_i} \right) \right. \\
&\quad \left. + \left( \max_y g(y, x_i) - \max_{y, z} \left( \frac{h^\top(y, z; 0)x_i + \eta(y, z)}{\sigma^2} \right) \right) \right\} - \log(|\mathcal{J}|) - \frac{M_2^2 - M_1^2}{2\sigma^2}.
\end{aligned}$$

As a consequence, we achieve the conclusion of part (b) of the theorem.

**PROOF OF THEOREM D.8** To achieve the conclusion of the theorem, we need the following uniform laws of large numbers results

$$\sup_{\theta} \left| \frac{1}{n} \sum_{i=1}^n \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0)x_i + \tau(\theta_{y, z}) \right) - \int \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0)x + \tau(\theta_{y, z}) \right) dP(x) \right| \rightarrow 0, \tag{50}$$

$$\begin{aligned}
L_n &= \sup_{\theta} \left| \frac{1}{n} \sum_{i=1}^n \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0)x_i + \tau(\theta_{y, z}) \right) \right\} \left( \frac{\|h(y', z'; 0)\|^2}{2} - \log \pi_{y'} \right) \right. \\
&\quad \left. \int \left( - \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0)x + \tau(\theta_{y, z}) \right) \right\} \left( \frac{\|h(y', z'; 0)\|^2}{2} - \log \pi_{y'} \right) \right) dP(x) \right| \rightarrow 0 \tag{51}
\end{aligned}$$

almost surely as  $n \rightarrow \infty$ . The proofs of these results are similar to those in Theorem B.6; therefore, they are omitted.

Going back to the original problem, we denote  $\theta^0 = \left( \{\mu^0(y)\}_{y=1}^K, \{\Gamma_0(l)\}_{l=1}^L, \{\pi_y^0\}_{y=1}^K, \{b_0(l)\}_{l=1}^L \right)$  the optimal solutions of population SPLD regularized K-means (Note that, the existence of these optimal solutions is guaranteed due to the compact assumptions with the parameter spaces  $\Theta_l$ ,  $\Omega$ , and  $\Xi_l$  for all  $1 \leq l \leq L$ ). Then, according to the formulation of SPLD regularized K-means, we will

have that

$$\begin{aligned}
V_n &\leq \frac{1}{n} \sum_{i=1}^n \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x_i^\top h^0(y, z; 0) + \tau(\theta_{y, z}^0) \right) \right\} \left( \frac{\|x_i - h^0(y', z'; 0)\|^2}{2} - \log(\pi_{y', z'}^0) \right) \\
&= \frac{1}{2n} \sum_{i=1}^n \|x_i\|^2 - \frac{1}{n} \sum_{i=1}^n \max_{(y, z) \in \mathcal{J}} \left( x_i^\top h^0(y, z; 0) + \tau(\theta_{y, z}^0) \right) \\
&\quad + \frac{1}{n} \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x_i^\top h^0(y, z; 0) + \tau(\theta_{y, z}^0) \right) \right\} \left( \frac{\|h^0(y', z'; 0)\|^2}{2} - \log \pi_{y'}^0 \right) \\
&\quad + \log \left( \sum_{y=1}^K \exp \left( \max_{z \in \mathcal{L}} \tau(\theta_{y, z}^0) + \log \pi_y^0 \right) \right)
\end{aligned}$$

where we define

$$\begin{aligned}
h^0(y, z; 0) &= \Lambda_0(z; 1) \dots \Lambda_0(z; 1) \mu^0(y), \\
\Lambda_0(z; \ell) &:= \sum_{p \in \mathcal{P}(\ell)} s(\ell, p) T(t; \ell, p) B(\ell, p) \Gamma_0(\ell, p), \\
p_{y', z'}^0 &= \exp \left( \tau(\theta_{y', z'}^0) + \log \pi_{y'}^0 \right) / \left( \sum_{y=1}^K \exp \left( \max_{z \in \mathcal{L}} \tau(\theta_{y, z}^0) + \log \pi_y^0 \right) \right)
\end{aligned}$$

for all  $1 \leq \ell \leq L$ . From the results with uniform laws of large numbers in (50) and (51), we obtain that

$$\begin{aligned}
&\frac{1}{n} \sum_{i=1}^n \max_{(y, z) \in \mathcal{J}} \left( x_i^\top h^0(y, z; 0) + \tau(\theta_{y, z}^0) \right) \rightarrow \int \max_{(y, z) \in \mathcal{J}} \left( x^\top h^0(y, z; 0) + \tau(\theta_{y, z}^0) \right) dP(x), \\
&\frac{1}{n} \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x_i^\top h^0(y, z; 0) + \tau(\theta_{y, z}^0) \right) \right\} \left( \frac{\|h^0(y', z'; 0)\|^2}{2} - \log \pi_{y'}^0 \right) \\
&\quad \rightarrow \int \left( \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x^\top h^0(y, z; 0) + \tau(\theta_{y, z}^0) \right) \right\} \left( \frac{\|h^0(y', z'; 0)\|^2}{2} - \log \pi_{y'}^0 \right) \right) dP(x)
\end{aligned}$$

almost surely as  $n \rightarrow \infty$ . The above results lead to

$$\frac{1}{n} \sum_{i=1}^n \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x_i^\top h^0(y, z; 0) + \tau(\theta_{y, z}^0) \right) \right\} \left( \frac{\|x_i - h^0(y', z'; 0)\|^2}{2} - \log(\pi_{y', z'}^0) \right) \rightarrow \bar{V}$$

almost surely as  $n \rightarrow \infty$ . Therefore, we have  $\lim_{n \rightarrow \infty} V_n \leq \bar{V}$  almost surely as  $n \rightarrow \infty$ . On the other

hand, with the results of uniform laws of large numbers in (50) and (51), we also have that

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n \max_{(y,z) \in \mathcal{J}} \left( x_i^\top \hat{h}(y, z; 0) + \tau(\hat{\theta}_{y', z'}) \right) &\rightarrow \int \max_{(y,z) \in \mathcal{J}} \left( x^\top \hat{h}(y, z; 0) + \tau(\hat{\theta}_{y', z'}) \right) dP(x), \\ \frac{1}{n} \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x_i^\top \hat{h}(y, z; 0) + \tau(\hat{\theta}_{y', z'}) \right) \right\} &\left( \frac{\|\hat{h}(y', z'; 0)\|^2}{2} - \log \hat{\pi}_{y'} \right) \\ &\rightarrow \int \left( \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x_i^\top \hat{h}(y, z; 0) + \tau(\hat{\theta}_{y', z'}) \right) \right\} \right) \left( \frac{\|\hat{h}(y', z'; 0)\|^2}{2} - \log \hat{\pi}_{y'} \right) dP(x) \end{aligned}$$

almost surely as  $n \rightarrow \infty$  where  $\hat{\theta} = \left( \{\hat{\mu}(y)\}_{y=1}^K, \{\hat{\Gamma}(\ell)\}_{\ell=1}^L, \{\hat{\pi}_y\}_{y=1}^K, \{\hat{b}(\ell)\}_{\ell=1}^L \right)$  is the optimal solution of SPLD regularized K-means and

$$\begin{aligned} \hat{h}(y, z; 0) &= \hat{\Lambda}(z; 1) \dots \hat{\Lambda}(z; L) \hat{\mu}(y), \\ \hat{\Lambda}(z; \ell) &= \sum_{p \in \mathcal{P}(\ell)} s(\ell, p) T(t; \ell, p) B(\ell, p) \Gamma(\ell, p), \\ \hat{p}(y', z') &= \exp \left( \tau(\hat{\theta}_{y', z'}) + \log \hat{\pi}_{y'} \right) / \left( \sum_{y=1}^K \exp \left( \max_{z \in \mathcal{L}} \tau(\hat{\theta}_{y', z'}) + \log \hat{\pi}_y \right) \right) \end{aligned}$$

for all  $1 \leq \ell \leq L$ . Hence,

$$V_n \rightarrow \int \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( x_i^\top \hat{h}(y, z; 0) + \tau(\hat{\theta}_{y', z'}) \right) \right\} \left( \frac{\|x - \hat{h}(y', z'; 0)\|^2}{2} - \log(\hat{p}(y', z')) \right) dP(x) \geq \bar{V}$$

almost surely as  $n \rightarrow \infty$ . As a consequence, we achieve  $V_n \rightarrow \bar{V}$  almost surely as  $n \rightarrow \infty$ . We reach the conclusion of the theorem.

**PROOF OF THEOREM D.9** The proof of the theorem is a straightforward application of the results with uniform laws of large numbers established in the proof of Theorem D.8. In particular, we denote the following set

$$\begin{aligned} \mathcal{F}_0(\epsilon) &= \left\{ (S, \{\pi_y\}, \{b(\ell)\}) : S = \left\{ h(y, z; 0) : (y, z) \in \mathcal{J} \right\}, \right. \\ &\text{and } \inf_{(S_0, \{\pi_y^0\}, \{b_0(\ell)\}) \in \mathcal{G}(\mathcal{F}_0)} \left\{ H(\hat{S}_n, S'_0) + \sum_{y=1}^K |\pi_y - \pi_y^0| + \sum_{z \in \mathcal{L}} \|b(\ell) - b_0(\ell)\| \right\} \geq \epsilon \left. \right\} \end{aligned}$$

for any  $\epsilon > 0$ . Since the parameter spaces of  $\theta$  are compact sets, the set  $\mathcal{F}_0(\epsilon)$  is also a compact set for all  $\epsilon > 0$ . Denote

$$\begin{aligned} g(S, \{\pi_y\}, \{b(\ell)\}) &= \int \sum_{(y', z') \in \mathcal{J}} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}} \left( h^\top(y, z; 0)x + \tau(\theta_{y, z}) \right) \right\} \\ &\quad \times \left( \frac{\|x - h(y', z'; 0)\|^2}{2} - \log(\pi_{y', z'}) \right) dP(x) \end{aligned}$$

for all  $(S, \{\pi_y\}, \{b(\ell)\})$ . From the definition of  $\mathcal{F}_0(\epsilon)$ , we have that

$$g(S, \{\pi_y\}, \{b(\ell)\}) > g(S_0, \{\pi_y^0\}, \{b_0(\ell)\})$$

for all  $(S, \{\pi_y\}, \{b(\ell)\}) \in \mathcal{F}_0(\epsilon)$  and  $(S_0, \{\pi_y^0\}, \{b_0(\ell)\}) \in \mathcal{G}(\mathcal{F}_0)$ . As  $\mathcal{F}_0(\epsilon)$  is a compact set, we further have that

$$\inf_{(S, \{\pi_y\}, \{b(\ell)\}) \in \mathcal{F}_0(\epsilon)} g(S, \{\pi_y\}, \{b(\ell)\}) > g(S_0, \{\pi_y^0\}, \{b_0(\ell)\})$$

for all  $(S_0, \{\pi_y^0\}, \{b_0(\ell)\}) \in \mathcal{G}(\mathcal{F}_0)$  and  $\epsilon > 0$ .

Now, according to the uniform laws of large numbers established in (50) and (51), we have that  $V_n \rightarrow g(\hat{S}_n, \{\hat{\pi}_y\}, \{\hat{b}(\ell)\})$  almost surely as  $n \rightarrow \infty$ . According to the result of Theorem D.8, it implies that  $g(\hat{S}_n, \{\hat{\pi}_y\}, \{\hat{b}(\ell)\}) \rightarrow g(S_0, \{\pi_y^0\}, \{b_0(\ell)\})$  almost surely for all  $(S_0, \{\pi_y^0\}, \{b_0(\ell)\}) \in \mathcal{G}(\mathcal{F}_0)$ . Therefore, for each  $\epsilon > 0$  we can find sufficiently large  $N$  such that we have

$$\inf_{(S_0, \{\pi_y^0\}, \{b_0(\ell)\}) \in \mathcal{G}(\mathcal{F}_0)} \left\{ H(\hat{S}_n, S_0') + \sum_{y=1}^K |\pi_y - \pi_y^0| + \sum_{z \in \mathcal{L}} \|b(\ell) - b_0(\ell)\| \right\} < \epsilon$$

almost surely for all  $n \geq N$ . As a consequence, we achieve the conclusion of the theorem.

**PROOF OF THEOREM D.10** The proof of this result relies on the evaluation of Rademacher complexity being established in the proof of Theorem B.6. In particular, from the definitions of  $V_n$  and  $\bar{V}_n$ , the conclusion of the theorem is equivalent to demonstrate that

$$\begin{aligned} A &= \left| \mathbb{E} \left\{ \frac{1}{n} \sum_{i=1}^n \sum_{(y', z') \in \bar{\mathcal{J}}_n} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \bar{\mathcal{J}}_n} \left( \hat{h}(y, z; 0)^\top x_i + \tau(\hat{\theta}_{y, z}) \right) \right\} \left( \frac{\|x_i - \hat{h}(y', z'; 0)\|^2}{2} - \log(\hat{p}(y', z')) \right) \right\} \right. \\ &\quad \left. - \mathbb{E} \left\{ \int \left( \sum_{(y', z') \in \bar{\mathcal{J}}_n} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \bar{\mathcal{J}}_n} \left( \hat{h}(y, z; 0)^\top x + \tau(\hat{\theta}_{y, z}) \right) \right\} \left( \frac{\|x_i - \hat{h}(y', z'; 0)\|^2}{2} - \log(\hat{p}(y', z')) \right) \right) dP(x) \right\} \right| \\ &\leq \frac{4\bar{c}_n |\mathcal{J}| R^2}{\sqrt{n}} + (2 \log(1/\bar{\gamma}) + R^2) \sqrt{\frac{2(\log 8 + (D^{(0)} + 1)(\bar{c}_n |\mathcal{J}| - 1) \log n)}{n}}. \end{aligned}$$

In fact, we can rewrite  $A$  as

$$\begin{aligned} A &= \left| \mathbb{E} \left\{ \frac{1}{n} \sum_{i=1}^n \sum_{(y', z') \in \bar{\mathcal{J}}_n} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \bar{\mathcal{J}}_n} \left( \hat{h}(y, z; 0)^\top x_i + \tau(\hat{\theta}_{y, z}) \right) \right\} \left( l_{\hat{h}(y', z'; 0)}(x_i) - \tau(\hat{\theta}_{y', z'}) - \log \hat{\pi}_{y'} \right) \right\} \right. \\ &\quad \left. - \mathbb{E} \left\{ \int \left( \sum_{(y', z') \in \bar{\mathcal{J}}_n} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \bar{\mathcal{J}}_n} \left( \hat{h}(y, z; 0)^\top x + \tau(\hat{\theta}_{y, z}) \right) \right\} \left( l_{\hat{h}(y', z'; 0)}(x) - \tau(\hat{\theta}_{y', z'}) - \log \hat{\pi}_{y'} \right) \right) dP(x) \right\} \right| \end{aligned}$$

where  $l_y(x) = -x^\top y + \|y\|^2/2$  for all  $(x, y)$ . By means of the standard symmetrization argument from empirical processes theory [61], we obtain that

$$\begin{aligned}
A &\leq \mathbb{E} \left| \sum_{i=1}^n \sum_{(y', z') \in \bar{\mathcal{J}}_n} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \bar{\mathcal{J}}_n} \left( \hat{h}(y, z; 0)^\top x_i + \tau(\hat{\theta}_{y, z}) \right) \right\} \left( l_{\hat{h}(y', z'; 0)}(x_i) - \tau(\hat{\theta}_{y', z'}) - \log \hat{\pi}_{y'} \right) \right. \\
&\quad \left. - \int \left( \sum_{(y', z') \in \bar{\mathcal{J}}_n} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \bar{\mathcal{J}}_n} \left( \hat{h}(y, z; 0)^\top x + \tau(\hat{\theta}_{y, z}) \right) \right\} \left( l_{\hat{h}(y', z'; 0)}(x) - \tau(\hat{\theta}_{y', z'}) - \log \hat{\pi}_{y'} \right) \right) dP(x) \right| \\
&\leq \mathbb{E} \sup_{\theta} \sup_{|\mathcal{J}'| \leq \bar{c}_n |\mathcal{J}|} \left| \frac{1}{n} \sum_{i=1}^n \sum_{(y', z') \in \mathcal{J}'} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}'} \left( h^\top(y, z; 0) x_i + \tau(\theta_{y, z}) \right) \right\} \right. \\
&\quad \left. \times \left( l_{h(y', z'; 0)}(x_i) - \tau(\theta_{y', z'}) - \log \pi_{y'} \right) \right. \\
&\quad \left. - \int \left( \sum_{(y', z') \in \mathcal{J}'} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}'} \left( h^\top(y, z; 0) x + \tau(\theta_{y, z}) \right) \right\} \left( l_{h(y', z'; 0)}(x) - \tau(\theta_{y', z'}) - \log \pi_{y'} \right) \right) dP(x) \right| \\
&\leq 2 \mathbb{E} \sup_{\theta} \sup_{|\mathcal{J}'| \leq \bar{c}_n |\mathcal{J}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \sum_{(y', z') \in \mathcal{J}'} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}'} \left( h^\top(y, z; 0) x_i + \tau(\theta_{y, z}) \right) \right\} \right. \\
&\quad \left. \times \left( l_{h(y', z'; 0)}(x_i) - \tau(\theta_{y', z'}) - \log \pi_{y'} \right) \right| \quad (52)
\end{aligned}$$

where  $\sigma_1, \dots, \sigma_n$  are i.i.d. Rademacher random variables, i.e.,  $\mathbb{P}(\sigma_i = -1) = \mathbb{P}(\sigma_i = 1) = 1/2$  for  $1 \leq i \leq n$ . By means of the triangle inequality and the additive property of supremum, we have the following inequality

$$\begin{aligned}
&\mathbb{E} \sup_{\theta} \sup_{|\mathcal{J}'| \leq \bar{c}_n |\mathcal{J}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \sum_{(y', z') \in \mathcal{J}'} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}'} \left( h^\top(y, z; 0) x_i + \tau(\theta_{y, z}) \right) \right\} \right. \\
&\quad \left. \times \left( l_{h(y', z'; 0)}(x_i) - \tau(\theta_{y', z'}) - \log \pi_{y'} \right) \right| \\
&\leq \mathbb{E} \sup_{\theta} \sup_{|\mathcal{J}'| \leq \bar{c}_n |\mathcal{J}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \sum_{(y', z') \in \mathcal{J}'} 1 \left\{ (y', z') = \arg \max_{(y, z) \in \mathcal{J}'} \left( h^\top(y, z; 0) x_i + \tau(\theta_{y, z}) \right) \right\} \left( \frac{\|h(y', z'; 0)\|^2}{2} - \log \pi_{y'} \right) \right| \\
&\quad + \mathbb{E} \sup_{\theta} \sup_{|\mathcal{J}'| \leq \bar{c}_n |\mathcal{J}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \max_{(y, z) \in \mathcal{J}'} \left( h^\top(y, z; 0) x_i + \tau(\theta_{y, z}) \right) \right|. \quad (53)
\end{aligned}$$

According to the property of Rademacher complexity and the Cauchy-Schwarz's inequality, we get

$$\begin{aligned}
& \mathbb{E} \sup_{\theta} \sup_{|\mathcal{J}'| \leq \bar{c}_n |\mathcal{J}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \max_{(y,z) \in \mathcal{J}'} \left( h^\top(y, z; 0) x_i + \tau(\theta_{y,z}) \right) \right| \\
& \leq \mathbb{E} \sup_{|S'| \leq \bar{c}_n |\mathcal{J}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \max_{s \in S'} s^\top [x_i, 1] \right| \\
& \leq 2\bar{c}_n |\mathcal{J}| \mathbb{E} \sup_{s \in \mathbb{B}(R)} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i s^\top [x_i, 1] \right| \\
& \leq 2\bar{c}_n |\mathcal{J}| R \mathbb{E} \left\| \frac{1}{n} \sum_{i=1}^n \sigma_i [x_i, 1] \right\| \leq \frac{2\bar{c}_n |\mathcal{J}| (R^2 + 1)}{\sqrt{n}}. \tag{54}
\end{aligned}$$

Additionally, using the results with VC dimension of Voronoi partitions in the proof of Theorem D.8, we achieve that

$$\begin{aligned}
& \mathbb{E} \sup_{\theta} \sup_{|\mathcal{J}'| \leq \bar{c}_n |\mathcal{J}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \sum_{(y', z') \in \mathcal{J}'} 1 \left\{ (y', z') = \arg \max_{(y,z) \in \mathcal{J}'} \left( h^\top(y, z; 0) x_i + \tau(\theta_{y,z}) \right) \right\} \left( \frac{\|h(y', z'; 0)\|^2}{2} - \log \pi_{y'} \right) \right| \\
& = \mathbb{E} \left( \mathbb{E}_{\sigma} \sup_{\theta} \sup_{|\mathcal{J}'| \leq \bar{c}_n |\mathcal{J}|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \sum_{(y', z') \in \mathcal{J}'} 1 \left\{ (y', z') = \arg \max_{(y,z) \in \mathcal{J}'} \left( h^\top(y, z; 0) x_i + \tau(\theta_{y,z}) \right) \right\} \right. \right. \\
& \quad \left. \left. \times \left( \frac{\|h(y', z'; 0)\|^2}{2} - \log \pi_{y'} \right) \right| x_1, \dots, x_n \right) \\
& \leq \left( \log(1/\bar{\gamma}) + R^2/2 \right) \mathbb{E} \left( \mathbb{E}_{\sigma} \sup_{|S'| \leq \bar{c}_n |\mathcal{J}|} \sum_{j=1}^{|S'|} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i 1 \left\{ j = \arg \min_{1 \leq l \leq |S'|} [x_i, 1]^\top s_l \right\} \right| x_1, \dots, x_n \right) \\
& \leq \left( \log(1/\bar{\gamma}) + R^2/2 \right) \sqrt{\frac{2(\log 8 + (D^{(0)} + 1)(\bar{c}_n |\mathcal{J}| - 1) \log n)}{n}}. \tag{55}
\end{aligned}$$

where the supremum in the above inequality is taken with  $S' = \{s_1, \dots, s_{|S'|}\}$  for all  $|\mathcal{J}'| \leq \bar{c}_n |\mathcal{J}|$ . By combining the results from (52) (53), (54), and (55), we achieve the conclusion of the theorem.

## F Appendix F

In this appendix, we provide detail descriptions for several simulation studies in the main text.

### F.1 Architecture of the Network Used in Our Semi-Supervised Experiments

Table 6: The network architecture used in all of semi-supervised experiments on CIFA10, CIFAR100 and SVHN.

NAME	DESCRIPTION
input	$32 \times 32$ RGB image
conv1a	128 filters, $3 \times 3$ , pad = 'same', LReLU ( $\alpha = 0.1$ )
conv1b	128 filters, $3 \times 3$ , pad = 'same', LReLU ( $\alpha = 0.1$ )
conv1c	128 filters, $3 \times 3$ , pad = 'same', LReLU ( $\alpha = 0.1$ )
pool1	Maxpool $2 \times 2$ pixels
drop1	Dropout, $p = 0.5$
conv2a	256 filters, $3 \times 3$ , pad = 'same', LReLU ( $\alpha = 0.1$ )
conv2b	256 filters, $3 \times 3$ , pad = 'same', LReLU ( $\alpha = 0.1$ )
conv2c	256 filters, $3 \times 3$ , pad = 'same', LReLU ( $\alpha = 0.1$ )
pool2	Maxpool $2 \times 2$ pixels
drop2	Dropout, $p = 0.5$
conv3a	512 filters, $3 \times 3$ , pad = 'valid', LReLU ( $\alpha = 0.1$ )
conv3b	256 filters, $1 \times 1$ , LReLU ( $\alpha = 0.1$ )
conv3c	128 filters, $1 \times 1$ , LReLU ( $\alpha = 0.1$ )
pool3	Global average pool ( $6 \times 6 \rightarrow 1 \times 1$ pixels)
dense	Fully connected $128 \rightarrow 10$
output	Softmax

### F.2 Training Details

#### F.2.1 Semi-Supervised Learning Experiments on CIFAR10, CIFAR100, and SVHN

The training losses are discussed in Section 3.3. In addition to the cross-entropy loss, the reconstruction loss, and the RPN regularization, in order to further improve the performance of NRM, we introduce two new losses for training the model. Those two new losses are from our derivation of batch normalization using the NRM framework and from applying variational inference on the NRM. More details on these new training losses can be found in Appendix AB.4. We compare NRM with state-of-the-art methods on semi-supervised object classification tasks which use consistency regularization, such as the  $\Pi$  model [5], the Temporal Ensembling [5], the Mean Teacher [6], the Virtual Adversarial Training (VAT), and the Ladder Network [44]. We also compare NRM with methods that do not use consistency regularization including the improved GAN [7] and the Adversarially Learned Inference (ALI) [8].

All networks were trained using Adam with learning rate of 0.001 for the first 20 epochs. Adam momentum parameters were set to  $\text{beta1} = 0.9$  and  $\text{beta2} = 0.999$ . Then we used SGD with decayed

learning rate to train the networks for another 380 epochs. The starting learning rate for SGD is 0.15 and the end learning rate at epoch 400 is 0.0001. We use batch size 128. Let the weights for the cross-entropy loss, the reconstruction loss, the KL divergence loss, the moment matching loss, and the RPN regularization be  $\alpha_{CE}$ ,  $\alpha_{RC}$ ,  $\alpha_{KL}$ ,  $\alpha_{MM}$ , and  $\alpha_{PN}$ , respectively. In our training,  $\alpha_{CE} = 1.0$ ,  $\alpha_{RC} = 0.5$ ,  $\alpha_{KL} = 0.5$ ,  $\alpha_{MM} = 0.5$ , and  $\alpha_{PN} = 1.0$ . For Max-Min cross-entropy, we use  $\alpha^{max} = \alpha^{min} = 0.5$ .

### F.2.2 Supervised Learning Experiments with Max-Min Cross Entropy

**Training on CIFAR10** We use the 26 2 x 96d "Shake-Shake-Image" ResNet in [48] with the Cutout data augmentation suggested in [49] as our baseline. We implement the Max-Min cross-entropy on top of this baseline and turn it into a Max-Min network. In addition to Cutout data augmentation, standard translation and flipping data augmentation is applied on the 32 x 32 x 3 input image. Training procedure are the same as in [48]. In particular, the models were trained for 1800 epochs. The learning rate is initialized at 0.2 and is annealed using a Cosine function without restart (see [62]). We train our models on 1 GPU with a mini-batch size of 128.

**Training on CIFAR10** We use the Squeeze-and-Excitation ResNeXt-50 as in [50] as our baseline. As with CIFAR10, we implement the Max-Min cross-entropy for the baseline and turn it into a Max-Min network. During training, we follow standard practice and perform data augmentation with random-size cropping [63] to 224 x 224 x 3 pixels. We train the network with the Nesterov accelerated SGD for 125 epochs. The initial learning rate is 0.1 with momentum 0.9. We divide the learning rate by 10 at epoch 30, 60, 90, 95, 110, 120. Our network is trained on 8 GPUs with batch size of 32.

### F.2.3 Semi-Supervised Training on MNIST with 50K Labeled to Get the Trained Model for Generating Reconstructed Image in Figure 4

The architecture of the baseline CNN we use is given in the Table 7. We use the same training procedure as in Section F.2.1

Table 7: The network architecture used in our MNIST semi-supervised training.

NAME	DESCRIPTION
input	$28 \times 28$ image
conv1	32 filters, $5 \times 5$ , pad = 'full', ReLU
pool1	Maxpool $2 \times 2$ pixels
conv2a	64 filters, $3 \times 3$ , pad = 'valid', ReLU
conv2b	64 filters, $3 \times 3$ , pad = 'full', ReLU
pool2	Maxpool $2 \times 2$ pixels
conv3	128 filters, $3 \times 3$ , pad = 'valid', ReLU
pool3	Global average pool ( $6 \times 6 \rightarrow 1 \times 1$ pixels)
dense	Fully connected $128 \rightarrow 10$
output	Softmax